THEME ARTICLE: DATA MINING FOR CYBERSECURITY

Predicting Adversarial Cyber-Intrusion Stages Using Autoregressive Neural Networks

Aunshul Rege
Zoran Obradovic
Nima Asadi
Edward Parker
Rohan Pandit
Nicholas Masceri
Brian Singer
Temple University

Current cybersecurity approaches are response-driven and ineffective, as they do not account for dynamic adversarial movement. Using empirical evidence from observations of two Red Team–Blue Team cybersecurity training exercises held at Idaho National Laboratory and the Michigan Cyber Range, we used four different models to make temporal predictions of how adversaries progress through cyberattacks: nonlinear autoregressive (NAR) neural network, NAR neural network with exogenous input (NARX), NAR neural network for multi-steps-ahead prediction, and autoregressive integrated moving average (ARIMA). The obtained results demonstrate that the trained models can capture different variations in adversarial movement across the two datasets with reliable accuracy.

Rapidly advancing technologies have resulted in highly interconnected information networks and integrated systems, which has made them more accessible and vulnerable to cyberattacks.[1] Advanced persistent threats (APTs) have increasingly been targeting critical infrastructures such as power grids, transportation networks, and water supplies, constantly circumventing traditional reactive security measures and resulting in large and costly damages.[2] Organizations thus need to develop rapid, proactive, and dynamic defenses to more effectively manage APTs.[1] Many scholars have researched moving-target defense techniques that use spatiotemporal randomization to distort attackers' view of the network[3] as well as bipartite graph-based machine learning algorithms and synthetic data learning methods that serve as proactive filter-based methods for network defense.[1] These important contributions aid in the paradigm shift from reactive to anticipatory cybersecurity, but they are technical in nature and minimize the significance of adversarial trajectories.

Toward this end, we sought to create a framework for predicting how adversaries move as cyberattacks progress. Applying various machine learning algorithms and four different predictive models, we examined the adversarial cyber-intrusion stages using datasets of qualitative observations conducted at two cybersecurity training events: a US Industrial Control Systems Computer Emergency Response Team (ICS-CERT) Red Team/Blue Team cybersecurity training exercise held at Idaho National Laboratory (INL) in 2014, and a force-on-force paintball exercise held at the 2015 North American International Cyber Summit (NAICS 2015). In this article, we first discuss the 12 stages of adversarial movement, known as the intrusion chain. We then outline our methodology of creating time-series data and describe the predictive models. Next, we present and review the prediction results. Finally, we evaluate the findings and discuss possible implications for the intrusion chain model, as well as the importance and temporal characteristics of certain intrusion stages.

INTRUSION-CHAIN MODEL FOR ADVERSARIAL ATTACK TRAJECTORIES

Intrusion-chain models capture the step-by-step process of cyberattacks. We use the 12-stage model proposed by Mike Cloppert[2] (see Figure 1), as it allows for

- thorough analysis of adversaries' progression through the stages of a cyberattack,
- a means to assess how much time adversaries spend on different stages,
- exploring the possibly iterative nature of the cyberattack process through its cyclical structure, and
- incorporating human behavior (stages 2 and 4).

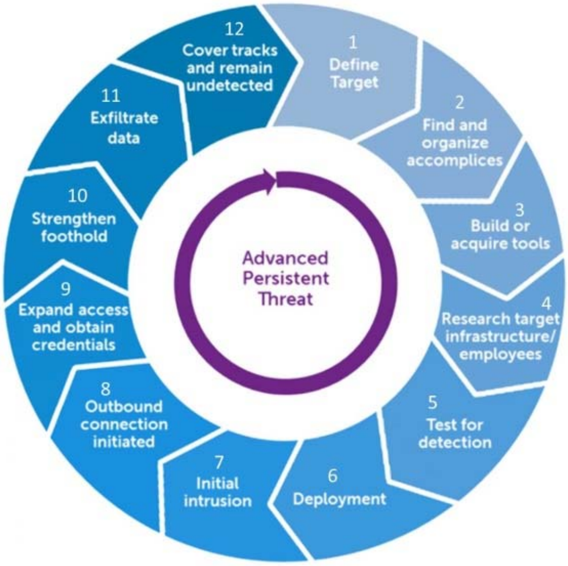
12
Cover tracks and remain undetected
1
Define Target
2
Find and organize accomplices
3
Build or acquire tools
4
Research target infrastructure/ employees
5
Test for detection
6
Deployment
7
Initial intrusion
8
Outbound connection initiated
9
Expand access and obtain credentials
10
Strengthen foothold
11
Exfiltrate data
Advanced Persistent Threat

Figure 1. Mike Cloppert's 12-stage intrusion-chain model.

The 12 stages are as follows. First, adversaries select their targets. Second, they find and form alliances with partners that complement and supplement their own skill sets. Adversaries then design and build their attack vectors and/or gather toolkits necessary to execute attacks. Fourth, they 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 in order to create appropriate evasion and response plans. Next, adversaries deploy their attack vectors, skills, and knowledge to gain a foothold into the target environment. In the seventh stage, they gain preliminary access to the targeted environment that allows them to install malware. Adversaries then establish more points of access into the targeted environment and penetrate additional systems that will increase their control. Tenth, adversaries who want to persist in their attacks will strengthen their presence by gaining credentials, using these to move laterally and deeper into the targeted environment and thereby establishing control over as many 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.

RESEARCH SITES FOR DATA COLLECTION

Red Team/Blue Team exercises (RTBTEs) are often used in the cybersecurity arena for training purposes and involve one group of security experts (Red Team) attacking a computer system while the opposing group (Blue Team) defends it.[4] 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 that system confidentiality, integrity, and availability are maintained during cyberattacks.[4] They 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. Our study collected data from two RTBTE research sites.

Research Site 1: ICS-CERT offers cybersecurity training events hosted at INL (henceforward referred to as ICS-CERT/INL). An 8-hour RTBTE was conducted in which the Red Team attacked cyber-physical systems that had to be defended by the Blue Team. The Red Team was randomly composed of 10 members who were a mix of system administrators, control systems engineers, and information technology specialists. Researchers observed the Red Team over the course of the exercise.

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 NAICS 2015, a force-on-force paintball exercise was conducted in which teams of four or five participants battled to penetrate and control Alphaville’s network, critical servers, and firewalls and then defend these controlled assets from rival teams. Researchers observed one of the competing teams that had four members. The exercise lasted 5 hours.

METHODOLOGY

At both sites, researchers conducted detailed observations, which formed the bulk of the dataset, and interviewed participants before and after the exercise to supplement and complement the 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 recorded each event, including the reactions and conversations of the team members as well as their time stamps. Then, they assigned each event to their matching intrusion stage from the 12-stage model. They used interviews conducted during and after the exercise to ensure that this was done correctly. The two datasets were then analyzed by transforming the observations from the compiled field notes into the time series representing the amount of time the team spent on each of the 12 intrusion-chain stages.

The time stamps of the start and end times of the events were used to generate the time series, where each time point in the time series represents 1-minute time span. Also, for each 1-minute time span, the value of the time series is determined by accumulating the time in minutes spent by the entire observed team on the corresponding intrusion stage. Figure 2 shows an example time series which was generated through this criteria for one intrusion stage (intrusion stage 6, deployment; see Figure 1) for the Red Team from the ICS-CERT/INL event. 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 1-minute time span. This figure shows that, for example, the observed Red Team spent 2 minutes on deployment stage during the one-minute time span between 7:41 am and 7:42 am.

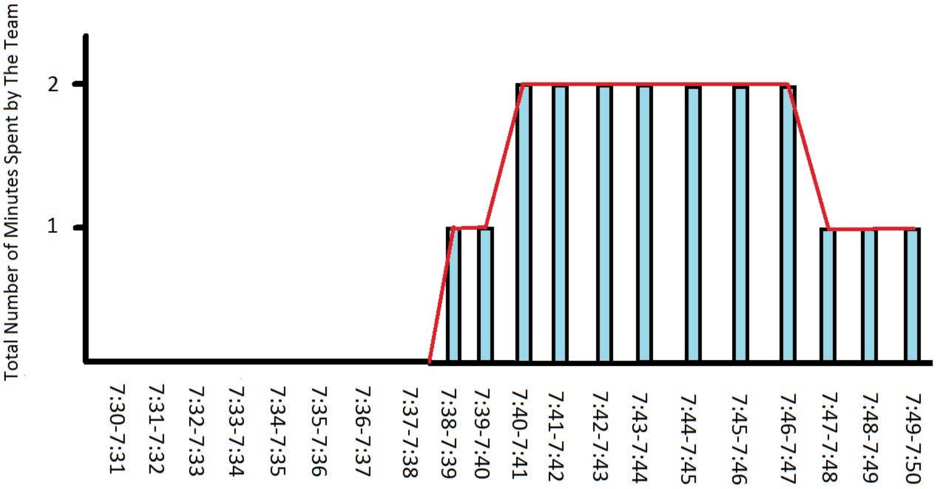

Figure 2. Example time series generated using the accumulated number of minutes spent by the entire Red Team for the first 20 minutes of intrusion stage 6 (Deployment) of the INL dataset within each 1-minute time interval.

After creating the time-series, several different structures of recurrent neural networks (RNNs) were used to predict the amplitudes of the future time points. RNNs are popular predictive models in identification and prediction of dynamic processes. Generally, this family of algorithms consists of a multi-layer perceptron (MLP) that takes as input a window of past independent inputs as well as past outputs to calculate 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 the inputs to the MLP. Therefore, if the prior activities of the attacking team members are recorded through a defense log mechanism, predicting their future activities becomes possible through this approach. In other words, the goal is to predict the amount of time the attacking team will invest on each intrusion-chain stage *n* steps later. This prediction can help improve 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.

PREDICTION MODELS

In this study we compare the results of four predictive models on our application: nonlinear autoregressive (NAR) neural network, NAR neural network with exogenous input (NARX), NAR neural network for multi-steps-ahead prediction, and autoregressive integrated moving average (ARIMA). Given our focus on the family of autoregressive models, other neural network models such as long short-term memory (LSTM) were not included.

Nonlinear Autoregressive Neural Network

In various applications, time series are characterized by high variations and sporadic behavior. This makes it difficult to model time series through linear approaches. Therefore, a nonlinear approach is more appropriate for analyzing such time series data. The amount of time the adversarial team spends on each intrusion stage can include fleeting transient periods for various reasons. These reasons include disruptions from the defense team, the attacking team's own failures, and changes in team dynamics. Therefore, we use nonlinear models to predict the values of the time-series data generated for the intrusion stages.

The first neural network structure employed in this study is the nonlinear autoregressive neural network (NAR). A NAR neural network model can be written as follows:[5]

$$y(t) = h(y(t_1), y(t_2), \ldots, y(t - p)) + e(t),$$

where the model is used to predict the value of a data series y at time t, hence $y(t)$, using the p past values of the series. Also, h is the network training function, and $e(t)$ is the error of approximation of the series y at time t. The function h is unknown in advance, and the training of the neural network aims to optimize the network weights and the neuron bias. The network training function in our study updates the weight and bias values according to Levenberg–Marquardt optimization procedure.[5]

NAR Neural Network with Exogenous Input

In many real applications, there is an important correlation between the modelled time series and additional external data. Thus, integrating knowledge or data about weather could benefit the time-series modelling process to provide an accurate forecast. The NAR neural network with exogenous or external input (NARX) proposed by Tsungnan Lin and his colleagues[6] predicts series given past values of series and another external series. The equation that models NARX network behavior for time-series prediction is as follows:[6,7]

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

As Figure 3 shows, the difference between this model and the NAR model is that in addition to the past values of the target time series, the input to NARX model includes the past values of another time series. We used two approaches to select the exogenous time series. In the first approach, to predict the value of each intrusion stage, we selected its previous stage according to the intrusion-chain model as the exogenous stage. For instance, to predict intrusion stage 6 (Deployment), we used the previous value of stage 5 (Test for Detection). In the second approach, we selected the most correlated time series to the target time series as the exogenous input. The reasoning behind this approach is that the most correlated time series provides the most information about the target time series, which can help increase the model's accuracy. We used Pearson's correlation coefficient formula to find correlations between the time series.

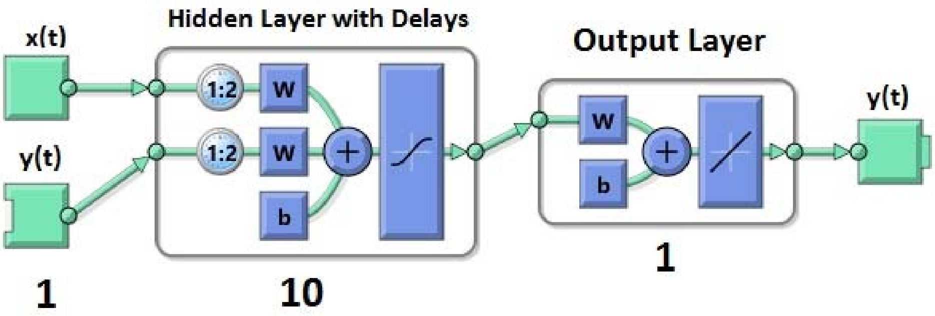

Figure 3. Nonlinear autoregressive neural network with exogenous input (NARX) model architecture.

NAR Neural Network for Multi-Steps-Ahead Prediction

In many applications, predicting the value of time series for more than one step ahead has significant value. For the purpose of predicting cyber-intrusion stages, it makes more robust proactive measures possible. We can train a neural network up to the present time point with all the known values of a time series in open-loop mode, then convert the architecture to closed-loop mode for multi-steps-ahead predictions of future values.[8] Thus, 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.

Autoregressive Integrated Moving Average

The autoregressive integrated moving average (ARIMA) model[9] predicts the values of time-series based on their prior values as well as the errors made by previous predictions. This allows the model to adjust to sudden variations 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 from the dependent variable and/or the lags from the prediction errors. This method can be modeled as:

$$x_t = \delta + \varphi_1 x_{t-1} + \varphi_2 x_{t-2} + w_t \,,$$

where x_t is a linear function of the values of x at the previous two times. Assuming 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 equation for making this prediction is as follows:

$$x_{p+1} = \delta + \varphi_1 x_p + \varphi_2 x_{p-1} + w_{p+1} \,.$$

In the equation above, the observed values of x_p and x_{p-1} are used and w_{p+1} is obtained as the assumed mean of the errors.

EXPERIMENTAL SETUP AND HYPERPARAMETERS

Each time point in the generated time series represents a 1-minute time span of the cyber training event. 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. Therefore, one time series representation is created for each intrusion stage. We created 10 time series for the first dataset and eight for the second dataset. The length of the generated time series for the first and second datasets were 480 and 300 minutes, respectively.

The datasets were randomly divided into two segments for training and testing purposes: 85 percent of each time series (336 minutes for dataset 1, and 210 minutes for dataset 2) was used for training, and 15 percent (72 minutes for dataset 1 and 45 minutes for dataset 2) for testing. Also, five-fold cross-validation was performed during training. This segmentation was applied to the two datasets in 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 had reliable accuracy, each model was trained 20 times, and the mean squared error (MSE) values were averaged to obtain a final MSE value.

PREDICTION RESULTS

Through the aforementioned experiments, we tried to predict the total time that the observed team would 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(s). We compared the models' predictive performance in terms of their error rate, namely the MSE, which measures the average of the squares of the errors and is defined as follows:

$$MSE = \sum_{t=1}^{n} \left(\hat{Y}_i - Y_i \right)^2 .$$

Figure 4 presents the prediction results of the NARX model for time series corresponding to two intrusion-chain time series (Deployment and Initial Intrusion) where the bottom subgraphs show the prediction error for each time point. The training (blue), validation (green), and testing (red) time points are randomly sampled. The error rate during each of these phases for the two time series is shown in the graphs on the right side of the figure. Note the convergence of the error rate as the neural network epoch progresses. The circles indicate the first epoch where the best (converged) performance is achieved. For instance, the best testing performance (red line) for Deployment is at epoch 4, where the MSE is 0.046.

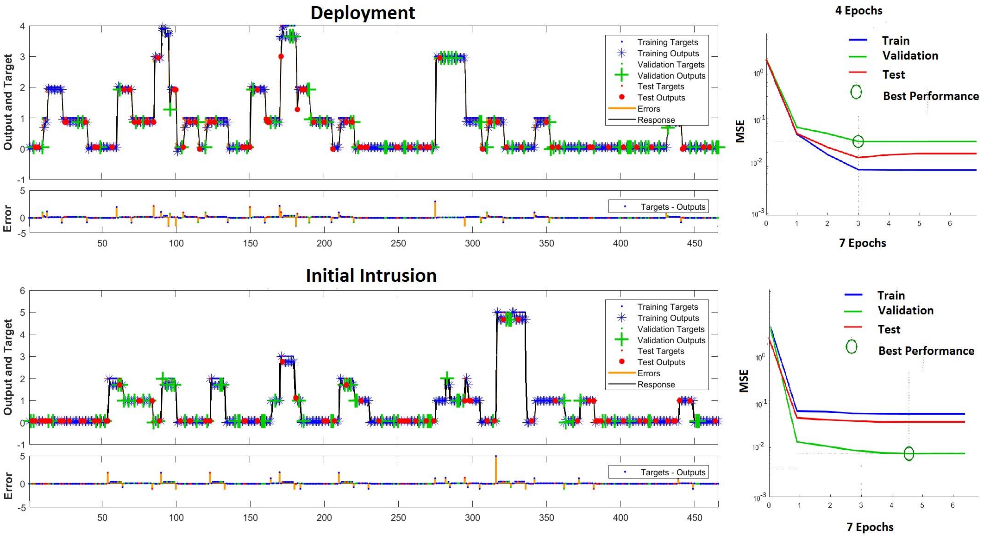

Figure 4. NARX prediction results for intrusion stages 6 (Deployment) and 7 (Initial Intrusion) for dataset 1.

The NARX method in this study included two variations of the exogenous input. The first approach used 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, to predict the value of a time point in the Initial Intrusion stage, in addition to its own time series, we use the Deployment stage 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 tried to find which time series could reinforce the neural network's learning power without considering the order of the time series based on the intrusion-chain model.

Figure 5 shows the correlation matrices for all intrusion stages for both datasets. For instance, in dataset 1 (left matrix), 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).

Tables 1 and 2 provide the complete prediction results for four neural network architectures: NAR; NARX in two variations, sequential (S) and correlation-based (C); ARIMA, and multi-steps-ahead NAR(M). Lower MSE results are displayed in bold. Because we did not observe intrusion stage 11 during the Alphaville experiment, it is not included in Table 2.

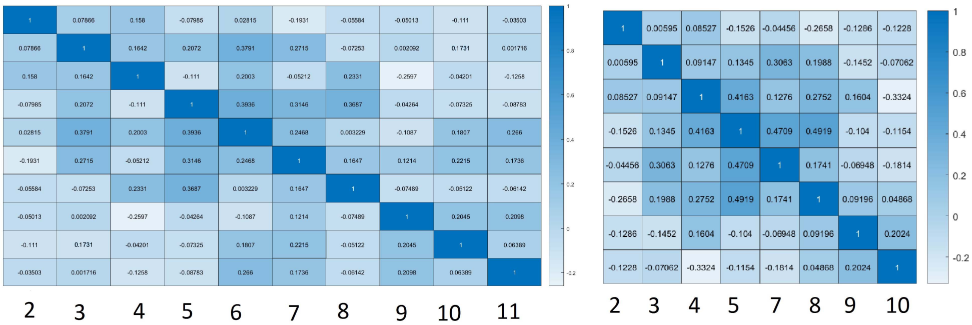

Figure 5. Correlation matrices for datasets 1 (left) and 2 (right). The numbers at the bottoms of the columns correspond to the intrusion stages in Figure 1.

Table 1. Comparison of the MSE values of the four prediction methods for dataset 1.

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. Comparison of the MSE values of the four prediction methods for dataset 2.

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

We can observe from Table 1 that NARX(C) provides the highest accuracy in six cases in the first experiment. These results offer empirical proof 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 intrusion-chain time series most relevant to the target time series and employ it to enhance the model's learning process. A similar conclusion can be drawn from the results of the second experiment in Table 2. In that experiment, the predictions made by incorporating the most correlated intrusion-stage time series show superior accuracy for more than half of the intrusion stages compared to the other models.

EVALUATION

This article offers a verifiable framework for dynamic prediction of adversarial movement across the cyber-intrusion chain. However, it is important to note that our analysis is based on two datasets, which has implications for the generalizability of the model prediction results. Many permutations and combinations of attack scenarios as well as different adversary types and motivations, objectives, and organizational dynamics cannot be accounted for by only two case studies. Therefore, more observations are needed to make a reliable mechanism for intrusion-chain analysis. Despite these limitations, our framework can be utilized in various experimental setups, 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.

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

NARX(C) was 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, with adversaries working on multiple intrusion chains (and stages) concurrently.

NARX(C) best predicted three stages of the intrusion chain: Find and Organize Accomplices (2), Expand Access and Obtain Credentials (9), and Strengthen Foothold (10). The stages used for predicting these stages in dataset 1 were Research Target Infrastructure/Employees (4), Exfiltrate Data (11), and Initial Intrusion (7); for dataset 2, stages 4, 10, and 9 were used for prediction. Thus, for example, stage 9 was used to predict stage 10. This meant that the intruders spent more time on these stages (relative to the other stages), for similar durations, at similar temporal points during the exercise.

Prior research[10] suggests that after intruders experience disruptions (caused by opponent players), they spend more time on certain stages for at least 5–10 minutes immediately after the disruptions. Therefore, we used a closed-loop architecture to predict the next 10 time steps of the intrusion stages using feedback from previous predictions. This offered insight into which stages intruders focused on, how much time they spent on these stages, and whether they also focused on other stages simultaneously. As Tables 1 and 2 show, our results suggest that a more accurate 1-minute prediction was achieved compared to the 10-minutes-ahead prediction using the closed-loop network. However, both the 1-minute and 10-minutes-ahead models predict adversarial movements within 1- and 10-minute time stamps, respectively, and show strong performance adapting to sudden variations in the process, which can help illuminate variations in attack progression.

CONCLUSION

Cybersecurity experts have identified five APT trends—there will be more attacks, more obfuscation, continued false attribution, greater shifts from opportunity-based attacks to more targeted attacks, and more damage ranging from data manipulation to data encryption or deletion. Governments and companies must continuously adapt to an ever-changing and evolving threat landscape by embracing proactive cybersecurity approaches that try to anticipate the who, what, where, when, and how of attacks and the best way to prepare for them. This article proposed an innovative mixed-methods-based approach to predict adversarial movement that our experimental results reveal to be reliable, though further research is needed to evaluate and compare different machine learning models.

ACKNOWLEDGMENTS

The material in this article is based upon work supported by a National Science Foundation CAREER Award, Grant No. 1453040, and partially by a 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 the 2015 North American International Cyber Summit.

REFERENCES

1. R. Colbaugh and K. Glass, *Proactive Defense for Evolving Cyber Threats*, report SAND2012-10177, Sandia National Laboratories, November 2012; https://fas.org/irp/eprint/proactive.pdf.
2. M. Cloppert, *Security Intelligence: Attacking the Cyber Kill Chain*, blog, 14 October 2009; http://digital-forensics.sans.org/blog/2009/10/14/security- intelligence-attacking-the-kill-chain.
3. J.H. Jafarian, E. Al-Shaer, and Q. Duan, “Spatio-temporal Address Mutation for Proactive Cyber Agility against Sophisticated Attackers,” *Proc. 1st ACM Workshop Moving Target Defense* (MTD 14), 2014, pp. 69–78.
4. D. Drinkwater and K. Zurkus, *Red Team versus Blue Team: How to Run an Effective Simulation*, CSO, 27 July 2016; https://www.csoonline.com/article/2122440/disaster-recovery/emergency-preparedness-red-team-versus-blue-team-how-to-run-an-effective-simulation.html.
5. T.W.S. Chow and C.T. Leung, “Non-linear Autoregressive Integrated Neural Network Model for Short-Term Load Forecasting,” *IEEE Proc.—Generation, Transmission, and Distribution*, vol. 143, no. 5, 1996, pp. 500–506.
6. T. Lin et al., “Learning Long-Term Dependencies in NARX Recurrent Neural Networks,” *IEEE Trans. Neural Networks*, vol. 7, no. 6, 1996, pp. 1329–1338.
7. H.T. Siegelmann, B.G. Horne, and C.L. Giles, “Computational Capabilities of Recurrent NARX Neural Networks,” *IEEE Trans. Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 27, no. 2, 1997, pp. 208–215.
8. A. Andalib and F. 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*, vol. 50, no. 3, 2009, pp. 739–747.
9. J. Rob and G. Athanasopoulos, *Notation for ARIMA Models Time Series Forecasting System*, SAS Inst..
10. A. Rege et al., “A Temporal Assessment of Cyber Intrusion Chains Using Multidisciplinary Frameworks and Methodologies,” *Proc. 2017 Int’l Conf. Cyber Situational Awareness, Data Analytics, and Assessment* (CyberSA 17), 2017; doi.org/10.1109/CyberSA.2017.8073398.

ABOUT THE AUTHORS

Aunshul Rege is an associate professor in the Department of Criminal Justice at Temple University. Her research examines cyberattacks/security from a human behavioral perspective, focusing on adversarial decision-making, adaptation to disruptions, and group dynamics. She received a PhD in criminal justice from the Rutgers School of Criminal Justice; she also has a BSc in computer science from the University of British Columbia. Contact her at rege@temple.edu.

Zoran Obradovic is the Laura H. Carnell Professor of Data Analytics, a professor in the Department of Computer and Information Sciences, and a professor in the Department of Statistical Science at Temple University, where he is also director of the Center for Data Analytics and Biomedical Informatics. In addition, Obradovic is an academician at Academia Europaea and a foreign academician at the Serbian Academy of Sciences and Arts. His research interests include data science and complex networks in decision support systems. He received a PhD in computer science from Pennsylvania State University. Contact him at zoran.obradovic@temple.edu.

Nima Asadi is a PhD student in the Department of Computer and Information Sciences at Temple University. His research interests include multivariate temporal data and network prediction and analysis. Contact him at nima.asadi@temple.edu.

Edward Parker was a PhD student in the Department of Criminal justice at Temple University. He is interested in national security and critical infrastructure protection. Contact him at ed.parker@temple.edu.

Rohan Pandit was an undergraduate student in the Department of Computer and Information Science at Temple University. His research interests include cybersecurity, artificial intelligence, machine learning and robotics. Contact him at rohan.pandit@temple.edu.

Nicholas Masceri was an undergraduate student in the Department of Criminal Justice at Temple University. His research interests include cybersecurity and human behavior. Contact him at nicholas.masceri@temple.edu.

Brian Singer was an undergraduate student in the Department of Criminal justice at Temple University. His research interests include cybercrime, national security, and terrorism risk methodology. Contact him at brian.singer@temple.edu.