Applying Machine Learning to Historical Cyber Attack Data to Predict and Prevent Future Attacks

Isabella Fumai

bellafumai@gmail.com

Abstract

Geopolitical tensions are high, with instability and uncertainty in multiple conflict zones which ripple around the world to countries that support the rule of law, those that do not, and those that are otherwise affected by the sprawling crises. While these conflicts are catastrophic, they are also convenient cover for state-sponsored actors to launch cyber attacks to damage critical infrastructure, undermine democratic institutions and values, and gather sensitive information to prepare for future attacks. State-sponsored actors have become highly sophisticated, with capabilities that leverage the power of technology to prey on governments and companies. Yet despite this sophistication, their attack patterns often have similar digital fingerprints. This reality creates the possibility that past attacks may provide clues for future ones. This paper, which is based on a machine learning algorithm, finds that historical data alone can be sufficient to predict the nature of future attacks with reasonably reliable accuracy. In particular, the severity of the previous attacks gave reliable insight into the severity of future attacks.

I. Introduction

Cyber attacks have been rapidly increasing for a variety of reasons, including the ways in which technological advancements have leveled the playing field and created increasingly more sophisticated attack vectors. However, all attacks fundamentally depend on the actors involved: the attacker and the target. Understanding who attacks who opens the door to the key questions of why and how.

In the cyber context, the attacker has the advantage with the element of surprise, with time to choose and surveil the target, and to plan when and how best to strike. While every government and company has cyber defenses, not all are created equal, and even the most prepared targets have vulnerabilities stemming from technology or the human element.

Against this background, mature governments and companies not only implement zero-trust security programs, but also adopt measures to mitigate the risks of attacks. One emerging best practice that has been championed throughout the cyber defense community is to publicly share attack details through a government-led program, like the Joint Cyber Defense Collaborative led by the U.S. Cybersecurity and Infrastructure Security Agency. This type of public-private cooperation creates a virtuous cycle through which governments and companies can become better prepared for future attacks.

Information sharing is not without risk, however. Disclosing a successful attack can have severe consequences, from legal liability to reputational damage and a lack of public trust (Libicki, 2020). However, significant disclosures continue to be made, as governments and companies rise together to improve the world's collective cyber defenses. This has led to publicly-available datasets that can be used to assess how prior cyber attacks may be able to predict the likelihood of future ones.

This paper applies supervised machine learning to historical cyber data in order to detect patterns and predict future attacks. In essence, this model views the situation as both a regression and classification problem. For simplicity, certain elements of the historical data were converted to numerical values to make it easier for the model to process, even though some of the information is categorical. This allowed for the model to output numerical values that correspond to key predictive parameters, such as the severity level of a future attack. Overall, this analysis revealed an outline of the major aspects of possible future attacks, which in theory could be used by governments and companies to become better prepared.

II. Background

Despite an explosion in cyber attacks, the amount and type of data available is limited, both as a result of being classified from a government's perspective and confidential from a

company's perspective. There is also a limited amount of publicly-available research to date that has attempted to use machine learning in the cyber context.

My initial research found that current methods of assessing foreign threats are often problematic due to a lack of documents and gaps in information. As a result, some have suggested this problem may be appropriate for machine learning (Rubenstein, date unknown). For example, a Stanford University study found that diplomatic documents could be analyzed using machine learning to identify discernible threats before actual events occur, even though none of the model types used approached a 100% accuracy rate (Katagiri and Min, 2015). That study, however, was prepared before the rise in cyber attacks in recent years, as well as the improved sharing within the information security community. These continuing trends will hopefully yield more and richer data over time, which should lead to increased confidence in the model output.

III. Dataset

I used the Dyadic Cyber Incident Dataset 2.0 (DCID 2.0) published by the Harvard Dataverse for this research (Valeriano, 2022). This open source data focuses on nation-state cyber operations by analyzing 433 known incidents between a pair of countries – the attacker (*State A*) and the target (*State B*) – labeled collectively with a numeric "*DyaidPair*." Other variables in the dataset include the name of the attack (*Name*), start date of the attack (*interactionStartDate*), end date of the attack (*interactionEndDate*), method of the attack classified as a number from 1 to 4.4 (*method*), target type ranked on a scale of 1 to 3 (*targetType*), and severity of the attack ranked on a scale from 1 to 5 (*severity*).

I prepared the data by first narrowing the DCID 2.0 dataset to the 47 interactions between the United States and Russia (using the corresponding *DyaidPair* value of 2365). I then focused only on the variables of attack severity and length, with the latter calculated and appended into the dataset by subtracting each attack end date from the start date (*daysLength*). These two features were chosen because: they both work with numerical data rather than categorical data, making it easier for the model; they require the model to learn how to deal with a range (with severity ranked from 1 to 5), which allows me to assess the accuracy across the range and observe any differences; and they seem most valuable for drawing conclusions that could be applied to real-world recommendations for how to respond to future attacks.

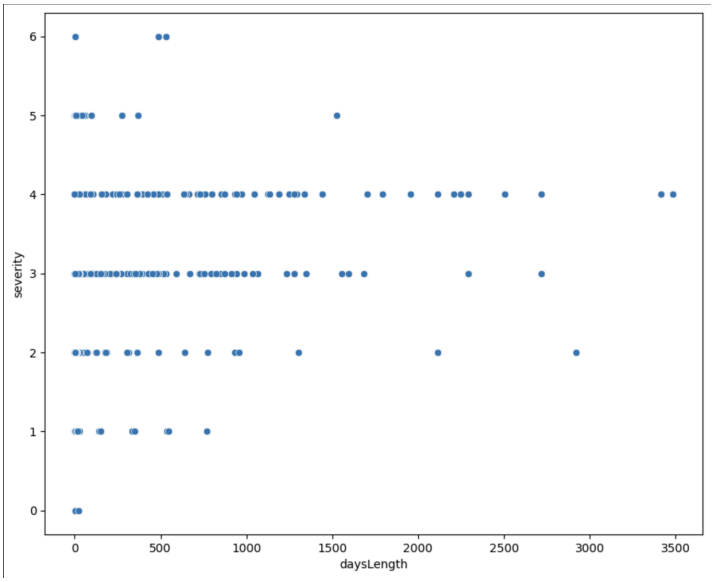

Figure 1: Scatterplot of *daysLength* vs. *severity*

I then split the 47 United States-Russia interactions into training and testing data, randomizing the interactions to allow the model to learn better. I chose to put approximately 80% of this randomized data to training and the remainder to testing.

IV. Methodology/Models

The methodology to create the model focused on the *daysLength* and *severity* values in the United States-Russia data. The model was given the code to use the ".append" feature that puts data in a list, which the model then applied to predict the next set of data in the list. For simplicity, the model was instructed to use the previous four cyber attacks to predict the next one, on the theory that some actors may attempt to launch similar attacks close in time.

```
US_Russia_a = df.loc[df['Dyadpair'] == 2365]
toDrop = ["interactionstartdate", "interactionenddate", "StateA", "StateB"]
US_Russia = US_Russia_a.drop(columns=toDrop)

raw_cols = US_Russia.columns

new_columns = []
prev_1_cols = []

for i in US_Russia.columns:
  new_columns.append([str(i + '_prev_1')])
  new_columns.append([str(i + '_prev_2')])
  new_columns.append([str(i + '_prev_3')])
  new_columns.append([str(i + '_prev_4')])


#US_Russia.columns
US_Russia[new_columns] = np.nan

US_Russia.head()

for i in range(len(US_Russia)):
  if i > 3:
    for col in raw_cols:
      US_Russia[str(col + '_prev_1')][i] = US_Russia[col][i-1]
      US_Russia[str(col + '_prev_2')][i] = US_Russia[col][i-2]
      US_Russia[str(col + '_prev_3')][i] = US_Russia[col][i-3]
      US_Russia[str(col + '_prev_4')][i] = US_Russia[col][i-4]
```

Figure 2: Screenshot of Data Illustrating the ".append" Feature

However, for the model to carry out its task, it needed access to the different machine learning algorithms. I gave access by importing the Python libraries and packages listed below.

```
%matplotlib inline
import warnings
import seaborn as sns
warnings.filterwarnings('ignore')
#import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.decomposition import PCA
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score,confusion_matrix, classification_report,roc_auc_score
from scipy.stats import zscore
from sklearn.model_selection import train_test_split
```

Figure 3: Screenshot of Model with Imported Libraries

I then created and used the Python-based machine learning algorithms discussed throughout this section. Each model was imported from sci-kit-learn, a free Python library.

Classification Using MLP Classifier

An Multilayer Perceptron Classifier (MLP Classifier) is a machine learning model used in cases with more complex data because it has multiple neurons and layers with different weights to account for some factors being more important or relevant than others. After importing this model from the sk.learn library, I set up the parameters, including the number of hidden layers that are processed before an output is produced. As mentioned previously, the data needed to be randomized to prevent the model from inappropriately noticing patterns that would frustrate the purpose of the research.

The model was then given the training data using the ".fit" method, which is a tool to instruct the model to become familiar with the data. After the model was ready, the next and last step was examining its accuracy in predicting the severity and length of the next cyber attack using the mean absolute error method. This test evaluates the average amount by which the prediction was off in comparison to the actual value it was trying to predict (*1.17. Neural,* n.d.).

```
from sklearn.neural_network import MLPClassifier

neural_model = MLPClassifier(
hidden_layer_sizes=(20),random_state=1, max_iter=300).fit(X_train, y_train)

neural_pred = neural_model.predict(X_test)

mean_absolute_error(y_test, neural_pred)
```

Figure 4: Screenshot of Code for Logistic Regression Model

Regression Using Logistic Regression Model

A Logistic Regression model is used to help calculate the probability of something, often for classification problems that involve categorical variables rather than numerical ones (*What is Logistic,* n.d.). Similar to the MLP Classifier, I imported it from the sk.learn library, exposed it

to the cyber attack data using the ".fit" method, and evaluated the output using the mean absolute error model.

```
from sklearn.linear_model import LogisticRegression
regression_model = LogisticRegression()
regression_model.fit(X_train, y_train)
```

```
▾ LogisticRegression
LogisticRegression()
```

```
predictions = regression_model.predict(X_test)
```

```
print(predictions)
print(y_test)

from sklearn.metrics import mean_absolute_error

mean_absolute_error(y_test, predictions)
```

Figure 5: Screenshot of Code for Logistic Regression Model

Regression Using Ridge Model

A Ridge model is a linear regression-based model that adjusts the categorical variables to be binary to help prevent overfitting (*What is ridge regression,* 2023). Applying this model to the relevant data required taking four steps: initialization (establishing a name for the model to use in the code); training (using the ".fit" method); prediction (having the model use the training data as an input to produce an output for the details regarding the next cyber attack); and evaluation (implementing the mean absolute error model).

```
from sklearn.linear_model import Ridge
import numpy as np
from sklearn.metrics import mean_absolute_error

# STEP 1: Initialization
ridge_model = Ridge(random_state=2)

# STEP 2: Training
ridge_model.fit(X_train, y_train)

# STEP 3: Prediction
predictions = ridge_model.predict(X_test)

# STEP 4: Evaluation
mean_absolute_error(y_test, predictions)
```

Figure 6: Screenshot of Code for Ridge Model

Classification and Regression Using Decision Tree Model

A Decision Tree model is used for classification and regression problems that have a hierarchy stemming from a base input that branches off to account for different possibilities and continues branching depending on if the inputted data satisfies the previous piece of data (*What is a Decision*, n.d.). Applying this model to the data required the same four steps used with the Ridge model: initialization, training, prediction, and evaluation.

```
from sklearn.model_selection import train_test_split

from sklearn.datasets import load_diabetes
from sklearn.model_selection import cross_val_score
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_absolute_error

# STEP 1: Initialization
decision_tree_model = DecisionTreeRegressor(max_depth=2, random_state=0)

# STEP 2: Training
decision_tree_model.fit(X_train, y_train)

# STEP 3: Prediction
predictions = decision_tree_model.predict(X_test)

# STEP 4: Evaluation
mean_absolute_error(y_test, predictions)
```

Figure 7: Screenshot of Code for Decision Tree Model

V. Results and Discussion

One core finding is that the models, when focused on the previous four cyber attacks in the historical data, were able to predict the severity and length of the next cyber attack with reasonable accuracy, especially when compared to only looking at one or two previous cyber attacks. This supports the belief that more data is better in this context, assuming similar quality and subject to eventual diminishing returns. That seems logical because the value of machine learning is unlocking the power of data to predict patterns that may otherwise be imperceptible to human analysis.

```
from sklearn.neural_network import MLPClassifier

neural_model = MLPClassifier(
hidden_layer_sizes=(850),random_state=1, max_iter=100).fit(X_train, y_train)

neural_pred = neural_model.predict(X_test)

mean_absolute_error(y_test, neural_pred)
```

```
0.8888888888888888
```

```
y = US_Russia['daysLength']
X = US_Russia.drop(columns=del_cols)


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

neural_model = MLPClassifier(
hidden_layer_sizes=(850),random_state=1, max_iter=100).fit(X_train, y_train)

neural_pred = neural_model.predict(X_test)

mean_absolute_error(y_test, neural_pred)
```

```
335.55555555555554
```

Figure 8: Screenshot of Code for MLP Classifier with the Predictions for *severity* on the Top and the Predictions for *daysLength* on the Bottom

Another finding is that each model was able to predict the severity of the next cyber attack with a much higher accuracy rate (relative to predictions on length). The likely explanation for this finding stems from the difference between the two variables. Specifically, the severity score could only range from 0 to 5, while the length of a cyber attack (measured in days) could run from a few days to multiple years, making the latter more random.

Additionally, using historical examples to predict length may be difficult because for longer attack lengths, the model is essentially making a prediction that an attack has already started and will only be detected after a certain number of days. In this sense, it may not be possible to fully track historical trends related to the length of attacks, as longer-length attacks would overlap with the very examples used to train the model. These difficulties can be

visualized through box plots and scatterplots to better understand the relationship between the severity and length variables.

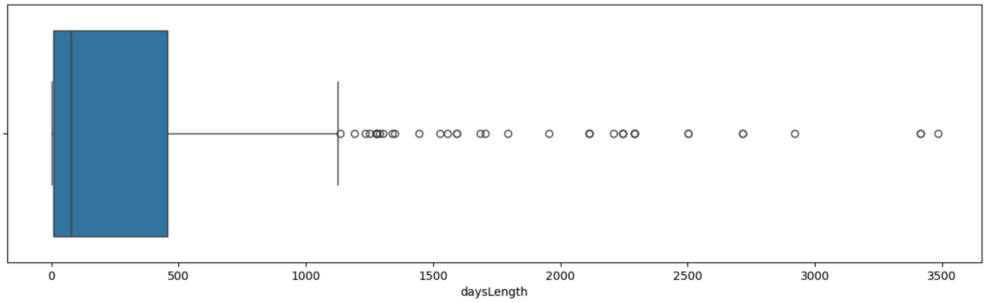

Figure 9: Box Plot of *daysLength*

After using the machine learning models listed above, two stood out: the MLP Classifier and the Ridge Model. The MLP Classifier model was the most accurate, which seems logical because the various neurons and layers helped account for the complexity of the cyber data. After realizing this, I decided to adjust the number of hidden layers and neurons in the model to see how it would impact the results. This eventually revealed a "sweet spot": either having more neurons in more layers (to create more predictive capacity), or fewer neurons in only one layer (to focus on a simpler prediction).

```
from sklearn.neural_network import MLPClassifier

neural_model = MLPClassifier(
hidden_layer_sizes=(20),random_state=1, max_iter=300).fit(X_train, y_train)

neural_pred = neural_model.predict(X_test)

mean_absolute_error(y_test, neural_pred)
```

```
0.4444444444444444
```

Figure 10: Screenshot of Results of One MLP Classifier with One Layer

On the other hand, the Ridge model stood out in a different way: it indicated an error in the initial processing of the data that needed to be fixed to move forward. The anomaly in the Ridge model involved the original accuracy being almost perfect with a mean absolute error value close to zero; however, looking more closely at the data revealed that the model concluded it could subtract the number of days from the *interactionStartDate* column (the start date of the cyber attack) from the *interactionEndDate column* (the end date of the attack) to find the real length of the next cyber attack. This result illustrates how machine learning models can adapt and use the training data to notice patterns in unexpected ways, in this case resulting in the model 'cheating' on its prediction. It also led me to adjust the logic and calibrate the data for future analysis.

```
from sklearn.linear_model import Ridge
import numpy as np
from sklearn.metrics import mean_absolute_error

# STEP 1: Initialization
ridge_model = Ridge(random_state=2)

# STEP 2: Training
ridge_model.fit(X_train, y_train)

# STEP 3: Prediction
predictions = ridge_model.predict(X_test)

# STEP 4: Evaluation
mean_absolute_error(y_test, predictions)
```

```
1.2445936990617357
```

Figure 11: Screenshot of Results of Ridge Model

Model accuracy was evaluated using the mean absolute error method, specifically to examine how far off the models' predictions were from the actual data it was trying to predict, with the lower value of the mean absolute error indicating the higher degree of output accuracy. Each time the code was run, the mean absolute error varied slightly due to the dataset being randomized after each trial to decrease the room for error stemming from having limited training data. This step was important, however, to minimize the risk of the model becoming too familiar with one particular set of data. The final MLP Classifier model was able to determine the severity, which ranges from 0 to 5, to within an average accuracy of 0.4 – meaning that when rounded, its predictive capability aligned to the correct value for that future attack. This helps prove the real-world value of using historical data to make predictions on severity.

VI. Conclusion

This paper applies machine learning to cybersecurity, specifically to assess whether historical, state-sponsored cyber attacks can help predict future ones. The models, especially the MLP Classifier, were reasonably accurate in predicting the severity of future cyber attacks – and to a lesser degree, the length of the attacks. These models, if refined and applied to more robust datasets, could be a tool to help governments and companies become better prepared for adversaries.

In particular, the United States government should continue to take a leading role in building a unified cyber defense between industry and government, by sharing details of known vulnerabilities and guidance on prevention and mitigation. This type of data would be invaluable in establishing a secure-by-default paradigm, and in better training a modern workforce to rise to the challenges of the cyber frontier.

To further this research, I intend to look for more diverse data, including less structured data that would need to be processed to handle more rigorous modeling and analysis. I also intend to run similar analyses across other countries to detect broader trends within the state-sponsored cyber communities.

Also, this research could be designed to focus on different signals in historical data to predict future attacks. For example, it may make sense to align the dataset to a chronological calendar of key events to give better context to why an attack may be launched at a certain time based on current global tensions. Similarly, more detail on the attack vectors may offer the type of digital clues that could lead to better diagnosis and attribution. For example, assuming numerical values to the various attack methods (e.g., ransomware = 1, phishing = 2, etc.) would create another predictive capability, presumably with a strong possibility for accuracy due to the low, controlled number of attack types. Regardless of how the data strategy is expanded and refined, based on the accuracy results of the models, the MLP Classifier still seems best equipped to process more complex data.

VII. Acknowledgments

I would like to thank my mentor, Simeon Sayer, and Inspirit AI for guiding me throughout this research process and helping me build and test out different machine learning models. Your help allowed me to perform this research and write this paper, and I am very appreciative.

VIII. References

Katagiri, Azusa and Min, Eric. *Identifying Threats: Using Machine Learning in International Relations*. (2015, September 15). Squarespace. Retrieved November 8, 2023, from https://static1.squarespace.com/static/5b429912f407b44e472b2118/t/5b42ad050e2e72bb806b681d/1531096326842/ThreatNoteDraft.pdf.

Libicki, M. (2020). *Cyberwar is What States Make of It.* The Cyber Defense Review, 5(2), 77–88. Retrieved November 8, 2023, from https://www.jstor.org/stable/26923524.

Maness, R. C., Valeriano, B., Hedgecock, K., Macias, J. M., & Jensen, B. (2023). *Expanding the Dyadic Cyber Incident and Campaign Dataset (DCID): Cyber Conflict from 2000 to 2020.* The Cyber Defense Review, 8(2), 65–90. Retrieved November 8, 2023, from https://www.jstor.org/stable/48743091.

Maness, R. C., Valeriano, B., Hedgecock, K., Macias, J. M., & Jensen, B. (2022, October 14). Tracking Competition in Cyberspace: Announcing the Dyadic Cyber Incident Dataset Version 2.0 - Modern War Institute. Modern War Institute -. Retrieved March 9, 2024, from https://mwi.westpoint.edu/tracking-competition-in-cyberspace-announcing-the-dyadic-cyber-incident-dataset-version-2-0/.

Neural network models (supervised) — scikit-learn 1.4.1 documentation. (n.d.). Scikit-learn. Retrieved February 19, 2024, from https://scikit-learn.org/stable/modules/neural_networks_supervised.html.

Perlroth, N. (2021, July 20). *How China Transformed Into a Prime Cyber Threat to the U.S. (Published 2021)*. The New York Times. Retrieved November 8, 2023, from https://www.nytimes.com/2021/07/19/technology/china-hacking-us.html.

Rubenstein, D. (n.d.). Nation-State Cyber Espionage and its Impacts. Retrieved November 8, 2023, from https://classes.engineering.wustl.edu/~jain/cse571-14/ftp/cyber_espionage/

Valeriano, B. (2022). *Dyadic Cyber Incident Database v 2.0*. Harvard Dataverse. Retrieved November 8, 2023, from https://dataverse.harvard.edu/file.xhtml?fileId=6503190&version=1.0.

Van Puyvelde, D. (2017, September 27). *Analysis | National security relies more and more on big data. Here's why.* The Washington Post. Retrieved November 8, 2023, from https://www.washingtonpost.com/news/monkey-cage/wp/2017/09/27/national-security-relies-more-and-more-on-big-data-heres-why/.

What is a Decision Tree. (n.d.). IBM. Retrieved February 19, 2024, from https://www.ibm.com/topics/decision-trees.

What is Logistic regression? (n.d.). IBM. Retrieved February 19, 2024, from https://www.ibm.com/topics/logistic-regression.

What is ridge regression? (2023, November 21). IBM. Retrieved February 19, 2024, from https://www.ibm.com/topics/ridge-regression.