Low-Interaction Honeypot Traffic: Modeling an Attacker’s Search

Samantha Roberts, M.S.

Nima Zahadat, Ph.D.

The George Washington University

**ABSTRACT**

Honeypots are a cybersecurity tactic used to divert attackers and analyze behavior patterns. The goal of this project was to classify and predict attack behaviors with data from a network of low-interaction honeypots. Chi-Square Goodness of Fit, Chi-Square Test of Independence, probability distribution function and a covariance matrix were conducted to determine statistical independence and significant interaction variables. Random Forest Classifier produced a 99% accuracy score on a sample size of destination ports. Feature importances were coalesced to identify two countries, two cities, and a few source Internet Protocol addresses as the most important variables to information gain. Hyperparameter tuning was applied to a sample of data for Logistic Regression, Support Vector Machine, and Multinomial Naïve Bayes models. The optimal model, Multinomial Naïve Bayes, with specified parameters was then conducted on the full data set with a resulting 95.3% accuracy score. Lastly, recommendations for further research and conclusions were drawn.

*Keywords:* honeypot, port-level attack, classification, machine learning, Random Forest Classifier, hyperparameter tuning, Multinomial Naïve Bayes

**INTRODUCTION**

Honeypots are a cybersecurity defensive tool used to deflect intrusive attempts and record attacker behaviors (Nawrocki, Wahlisch, Schmidt, Keil, & Schonfelderz, 2016). Two common subtypes available are low-interaction and high-interaction honeypots. These variations provide different experiences for businesses and attackers. High-interaction honeypots are capable of simulating response traffic to incoming requests for a more accurate representation. Low-interaction honeypots may present a façade architecture and record incoming network traffic, but they lack dynamic responsiveness. As a result, they do not capture the same complexity of transactions as high-interaction honeypots. This increases the challenge associated with pattern analysis and threat detection.

Cybersecurity is a critical part of the architecture of most companies and organizations in the modern age. There is no business or industry that is solely victimized by cyber-attacks and exploitation. As such, cybersecurity and defense strategies can be useful to all entities (Dua & Du, 2016). Despite limited interactivity, low-interaction honeypots are a popular option for many businesses in need of an improved, holistic cybersecurity strategy. In the event a business is unable to fund a robust honeypot network solution, economical options are available to create simulated low-interaction honeypots. One popular example is HoneyD, a free and open source software capable of simulating thousands of hosts, common network topology functions, and arbitrary operating system actions (Nawrocki et al., 2016). Much of the data collected through a low-interaction honeypot source is categorical in nature and often includes source and destination port names, timestamp of the transaction, the source’s Internet Protocol (IP) address, and the type of request traffic sent (e.g., session connection, data request).

The goal of this project was to classify and predict attack behaviors with data aggregated from a network of low-interaction honeypots. Significant behavior signatures were tested using hyperparameter tuning methods and a variety of classification models common to the cybersecurity realm. The results could be used to identify weak spots or commonly targeted attributes within a company’s structure, set action boundaries for intrusion prevention systems (IPS), and strengthen firewall rules; all depending upon whether the honeypot is deployed as an internet-facing structure or behind firewalls as a defense-in-depth strategy (Dua & Du, 2016; Prasad & Abraham, 2010). A successful outcome could inform businesses on where to harden their networks without investing in a resource-intensive solution.

**LITERATURE REVIEW**

Cybersecurity intrusion detection systems have been a focal point for several machine learning and data mining techniques. The technology used in this field often gather many data points and provide a plethora of data types, variables, and trends for analysis. An abundance of data is important for effective implementation of machine learning techniques; however, problems begin to arise in this field due to the velocity of changing data, which requires constant re-training of models and excessive data preprocessing to create usable data sets (Buczak & Guven, 2016). As a result, approaches to cybersecurity research and assessments should anticipate dedicated computer resources and personnel to the maintenance of any operational models.

The analysis of this project aligns most directly with misuse-based techniques to identify known trends within the network traffic. This approach tends to have a lower false positive rate but can perform poorly in identifying novel attacks. An ideal approach would incorporate considerations toward both misuse-based and anomaly detection by using a hybrid technique as suggested by Buczak and Guven (2016). In line with these researchers’ review, a supervised learning, multiclass classification approach was utilized for this project. This approach includes evaluation of the overall accuracy score and class detection rates, including true and false positives as exemplified in a classification report. The report visualizes precision, recall, and F1 scoring based upon the classification rates described.

Proper analysis of this data set required an in-depth review of similar classification and machine learning techniques related to honeypot network traffic. The first model was Logistic Regression, a linear classification model based on the sigmoid logit function. During the review, this model appeared the least often in cybersecurity and honeypot traffic analysis; however, it is a popular model due to its simplicity and reliability, so it was selected as a novel approach to this type of classification problem. One cybersecurity-related implementation of this model was the evaluation of SSH attacks as a comparison with SVM and Naïve Bayes models (Sadasivam, Hota, & Anand, 2016).

A variety of studies discussed by Buczak and Guyen indicated that the Random Forest Classifier could be accurate and particularly useful for knowledge gained about the variables involved (2016). It was cautioned that the greater the tree size, the more it would skew favor towards variables with more categories. Strengths included decreasing variance as the number of trees increases, and it is generally resistant to overfitting problems. Decision trees have also performed better than K-Nearest Neighbors (KNN) and some neural networks when incorporating a combination of performance metrics such as training time, accuracy, and rate of false positives and false negatives (Nawrocki et al., 2016).

Another model that was particularly popular in the literature was Support Vector Machine (SVM). This model is particularly well-suited to address many variables, which may not have robust samples within each sub-category. Additionally, it has been implemented successfully with data rebalancing techniques to undersample honeypot data sets (Diao, L., Yang, C., & Wang, 2012). The basis of the model is a binary classification between two groups but can be repeated in a one-versus-all strategy to generalize to multi-class problems (Buczak & Guyen, 2016). It often outperforms fellow models in training time performance and has been included in several honeypot-specific studies for misuse/signature detection (Goseva-Popstojanova et al., 2012; Diao; Dua & Du, 2016).

The final model reviewed for this project was the Multinomial Naïve Bayes Classifier. The model’s functionality is based on its calculation of the probabilities for each category and selecting the factor with the highest probability (Sadasivam et al., 2016). This model has historically performed accurately, with one main hindrance being a higher false positive rate as calculated by false positives divided by the sum of true negatives and false positives (Buczak & Guyen, 2016). In a system to implement dynamic firewall rules, this could lead to inappropriate blocking and denials of friendly users. Due to their nature, honeypots are naturally resistant to false positives in terms of data collection, which may assist in limiting any compounding effect that this classifier’s false positive rate deficiencies may present (Nawrocki et al., 2016).

**RESEARCH METHODOLOGY**

Initial data preprocessing and refinement activities were conducted using Python 3.6 (packages included JSON, codecs, CSV), CSV manipulation, and SQL Server Management Studio (SSMS). Java Script Object Notation (JSON) files were read into Python, the data was restructured, and saved into CSV files for ingestion into SSMS. Data cleaning efforts, including the removal of null values and data entry faults, were conducted in CSV and SSMS. The finalized data was read from SSMS into Python 3.6 to complete exploratory data analysis (EDA), statistical testing, and modeling efforts using packages including pandas, numpy, scipy, sklearn, and scikit-learn. A full list of packages required can be found in the README.md file in the project’s designated GitHub repository. Several tests and visualizations were chosen to identify significant variables and relationships, including Chi-Square tests, a probability distribution function, and Random Forest Classifier. Logistic Regression, SVM, and Multinomial Naïve Bayes models were used in a hyperparameter tuning sequence to determine an optimal model and associated parameters. The resulting model was then trained and tested on the full data set. Visualization materials for analysis and presentation were created using Python 3.6 (packages included matplotlib, seaborn) and Tableau. The associated website and presentation were developed with Hypertext Markup Language (HTML), Cascading Style Sheet (CSS), Java Script (JS), and hosted on GitHub pages. All associated project materials and code are publicly available at <https://www.github.com/samroberts58/Low-Interaction-Honeypot-Traffic>.

**DATA**

The main data for this project was located on Rapid7’s website as an open source data set titled “Rapid7 Heisenberg Cloud Honeypot cowrie Logs” (Rapid7, 2018). The initial data set included 13 JSON files with ten variables and contained approximately 12 million total records. According to information provided on Rapid7’s website, this data set has been used in combination with other data sources under the organization’s purview for analysis (Rapid7, 2018). As such, all other Rapid7 data available has purposely been excluded from this project to avoid duplicative efforts. Original variables including username, password, source port, and duration were removed due to few values available and complications with hidden character corruption.

Two additions to the original data set were the variables country and city. These were selected due to significant trends noted in other statistical modeling conducted on honeypots (Kaaniche, Deswarte, Alata, Dacier, & Nicomette, 2007). This data was assembled by identifying all unique source IP addresses in the data set during the data preprocessing stage. A bulk IP address locator website was used to identify geolocation data (Ip2geo Lookup, n.d.). Locations were matched to the source IP addresses in the original data set using a CSV VLOOKUP function. Any records that did not have a country value were removed. Any record values that did not have a city were assigned as “N/A” to retain a higher number of records. A brief description of the variables included in this project can be found in Appendix A.

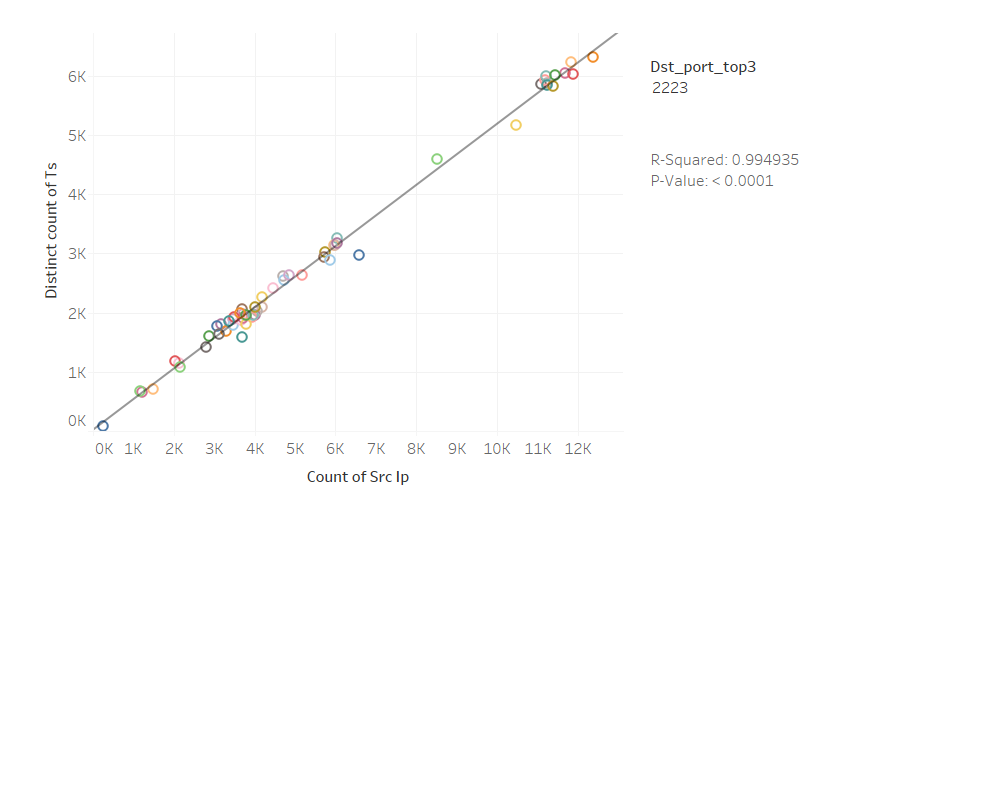
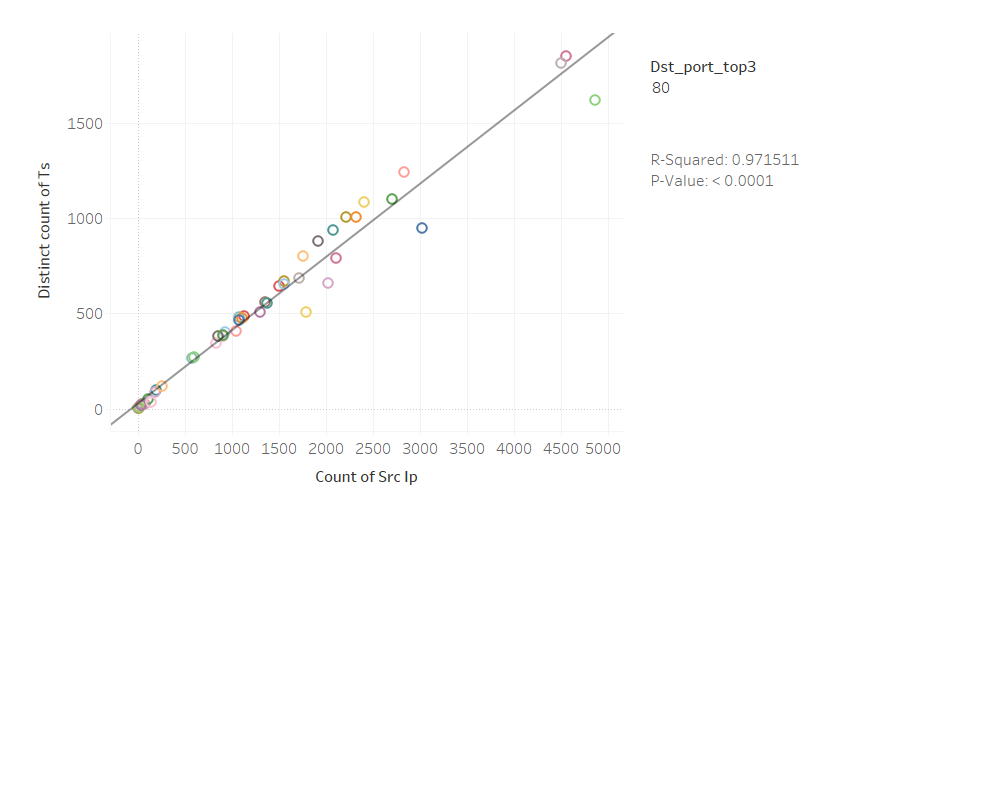
**DATA ANALYSIS**

Multiple columns were removed through the data preprocessing stage due to many missing values, corrupt hidden characters, and severely imbalanced category values. This reduced the total size of the data from 12 variables with 10,933,525 rows to 9 variables with 5,895,631 rows. During the review in SSMS, it was discovered the Netherlands accounted for 5,042,650 of the remaining rows or 47% of the overall data set before data cleansing efforts. Due to the severe skewness of the country, the Netherlands was removed from consideration. This provided the final 852,981 rows for analysis. Initial visualizations during the EDA phase included bar charts, pie charts, scatterplots, and an interaction plot. A sample size of 1,000 was chosen to ensure data distribution shape was possible to view in Python. Tableau was able to display the entire data set without issue. Statistical tests were conducted to determine whether the variable distributions were normal, verify the independence between variables, and identify variable aspects with higher probabilistic impact. A random sampling calculator was used to determine an ideal sample size of 9,497 for the given data set size at 95% confidence with 1% margin of error for all statistical tests conducted. Two additional variables, timestamp and session, were dropped due to p-values above 0.05 on the Chi-Square Test of Independence. This removed the ability to use time-series or long-range dependency models for a more complex analytical response (Zhan, Xu, & Xu, 2013). The Random Forest Classifier model was chosen as a preliminary classification model to determine whether the data could be successfully evaluated using supervised machine learning algorithms. The feature importances were extracted from the results and reviewed to determine variable details that provide the highest levels of information gain. Next, hyperparameter tuning methods were used to identify an ideal model and associated parameters for the following list: Logistic Regression, SVM, and Multinomial Naïve Bayes. Lastly, the most accurate model, Multinomial Naïve Bayes, was ran in batches on the full data set.

**KEY FINDINGS**

The EDA process generated several visualizations of the singular variables and some interaction plots. The purpose of this section was to identify clear trends and large outliers present in the data. It also became useful in setting initial expectations for underlying relationships between the variables. The single-variable bar and pie charts unanimously demonstrated skewed distributions, often with one or two subcategories with many records and all other subcategories with few results. Although many variables were skewed, none were nearly as impactful as the Netherlands and thus, were retained for further analysis. Additionally, the port-level analysis was quickly selected as the project’s focus due to lack of network or victim-level data, as discussed by Zhan et al. (2013).

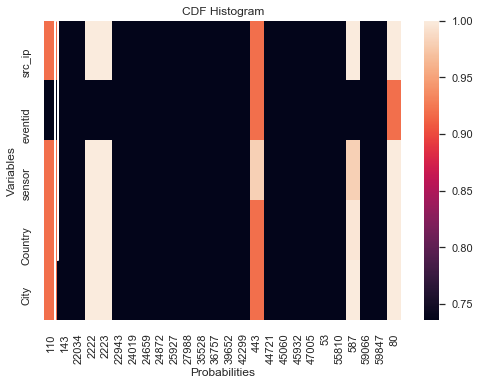
Scatterplots were created to visualize the interactions between two or more variables. A gentle curving trend was noted in plots that displayed the dynamic between countries and country attack rates, as derived by the count of country traffic divided by the count of distinct time values. There was also a significant relationship between destination ports and sensors, with R-squared ranging as high as 0.995 for port 2223 and 0.972 for port 80. Similar results were produced between the destination port and country as well. This supported the potential success of classification models as a structure for evaluation and prediction. Figures 1 and 2 show the scatterplots for port 2223 and port 80.



*Figure 1.* Attack traffic by time count & sensor, port 80. *Figure 2.* Attack traffic by time count & sensor, port 2223.

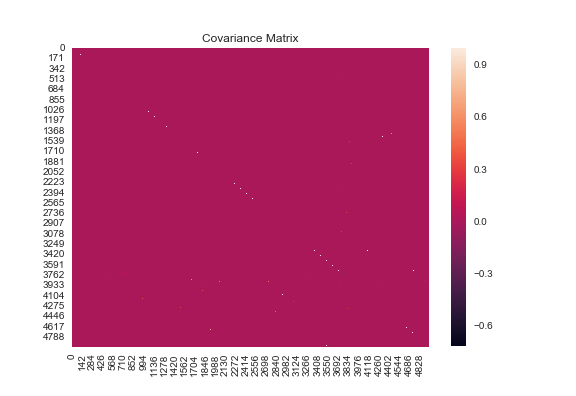
Statistical testing was a natural follow on to the EDA visualizations to verify the trends were significant. A sample size of 9,497 was applied to all the statistical tests completed during this project. Chi-Square Goodness of Fit tests were done on each variable individually. The null hypothesis was stated as the variables’ distributions were not normally distributed amongst their internal categories. None of the p-values returned as less than 0.05 so the results failed to reject the null hypothesis. Chi-Square Tests of Independence were conducted between each combination of variables to verify that there were statistically significant differences. Six of the eight variables were able to reject the null hypothesis when tested against each of the other seven variables. The variables timestamp and session were unable to reject the null hypothesis. Multiple models chosen for this project maintain an underlying assumption of independence. For this reason, these two variables were dropped from the data set and excluded from further analysis.

A probability distribution function was conducted on the remaining variables in relation to destination ports as the dependent variable. Five destination ports had significantly higher probabilities with most of the other variables. Event ID was exceptionally low across all destination ports, indicating less emphasis would be placed on this variable as a significant interactive factor in further modeling phases. It still retained practical value in the overall analysis as this variable may indicate a denial of service attack if severely heightened numbers of connection requests were flooding a sensor within a short period. Strong probabilities approached 1.0 for destination ports 110, 2222, 2223, 587, and 80 across multiple variables. Figure 3 shows the associated graph.



*Figure 3*. Probability Distribution Function

The final statistical evaluation was a covariance matrix. It was created to address multicollinearity in preparation for the Logistic Regression and Multinomial Naïve Bayes models. This required one hot encoding and label encoding actions against both the X and y sets before being split into training and testing samples. X\_train and X\_test were then scaled in preparation for numpy’s covariance matrix. The resulting matrix maintained almost all values close to zero, indicating no multicollinearity between the variables on a minute, categorical basis. The visualization can be seen in Figure 4.

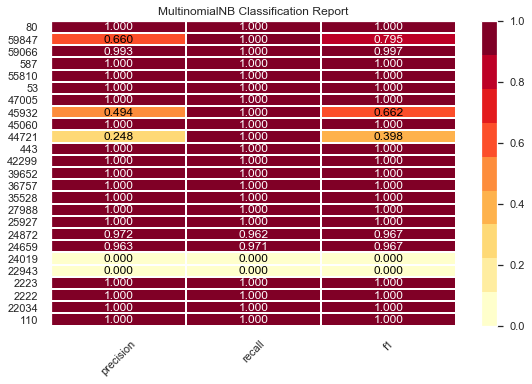


*Figure 4*. Covariance Matrix

The Random Forest Classifier was the first classification model conducted using the sample size previously assigned in the statistical testing phase. This was chosen as the beginning model in order to assess the data’s underlying relationships from yet another distinct perspective. The successful performance of this model also provided support for selecting additional classification algorithms to train and test the data set. Furthermore, it was able to rank and display significant categories as it pertained to information gain. The model performed at 99% accuracy when oversampling and 10-fold cross validation scoring techniques were implemented. A clipped tree (max depth = 5) image was created with top categories held by two countries, two cities, and a few specific source IP addresses. The two cities listed were not within the countries identified on the list, which indicated city could be an active variable in classifications independent of the country variable.

All the final models chosen for this project are supervised learning and classification-specific models. Logistic Regression was chosen as a multiclass linear classification method, which is based on the sigmoid logit function and corresponding probabilities. An underlying assumption of the model is that the variables are statistically independent, as was confirmed by the covariance matrix. SVM is also a linear classifier; however, this model’s core function is based on finding the maximum Euclidean distance between two classes’ closest points and is non-probabilistic in nature. The final model chosen, Multinomial Naïve Bayes, is a probability-based model that assumes the condition of independence between data points. The target variable selected for testing was destination port due to the practical use of analytical results and qualifying statistical testing results. Hyperparameter tuning was implemented to determine an ideal model with best-fit parameters from a variety of options. Two samples were randomly obtained from the main data set of sizes 100 and 1,000. Logistic Regression, SVM, and Multinomial Naïve Bayes models were selected with individually appropriate parameters. These three models were oversampled, fit, and cross-validated for comparison. Both sample sizes returned the same results and the best set of algorithm and parameters were automatically passed forward to the final model assessment.

The best model was Multinomial Naïve Bayes with alpha set at 0.01 and fit prior set as true. The hyperparameter tuning result estimated a 92% accuracy rate of the samples. Batch sampling sizes of 15,000 and 1,000 were implemented to iterate over the data set due to memory limitations. Upon each loop, a sample is randomly selected from the full data set, loaded into the sample data frame for processing, and subsequently deleted from the original data frame. Additionally, oversampling and cross-validation techniques were implemented to improve the accuracy of the model’s assessment. The two samples sizes returned accuracy ratings of 94.11% and 95.3%. A confusion matrix, classification report, and AUC score were generated for the results. The confusion matrix demonstrated five categories that were incorrectly classified and one category that was split partially between correct and incorrect classification. The falsely predicted categories were consistent within one class, albeit incorrectly assigned. The classification report continued to illuminate upon the results by addressing the precision, recall, and F1 score for each of the destination ports. Of the incorrectly classified categories, four had only two records per port, and the fifth had six records. The low record count for certain categories has been described by other researchers as one of two general causes of misclassifications (Goseva-Popstojanova, Anastasovski, & Pantev, 2012). This is a realistic side-effect of how honeypots operate due to the inability to funnel traffic to various destination ports equally. A significant majority of the destination ports achieved F1 scores at 0.967 or above, indicating both highly scored precision and recall. The AUC score calculated was 0.958, which supported the high accuracy of classification from the Multinomial Naïve Bayes model. Figure 5 displays the classification report.



*Figure 5*. Classification Report

**RECOMMENDATIONS**

There are several ways to improve upon the findings of this project. One way to elaborate on the project's goal would be to include additional data points. Location elements scored very highly with the Random Forest Classifier's feature importances function. Therefore, it could be beneficial to incorporate publicly-available information, like country holidays or significant political events, to identify a relationship with broader social and cultural influences and attack parameters. This type of information could be useful when addressing a company’s significant events as well. For example, announcing the release of a new product, opening a new store location, or changing leadership could attract additional attention from attackers. The event-based analysis could introduce additional dimensionality that would encourage longer-term trend analysis.

Another recommendation would be to create a customized recurrent neural network to find the most accurate model possible. This data set's features were linearly separable and able to be accurately classified by the Multinomial Naïve Bayes model and would thus make a good candidate for extended deep learning analysis. Previous research also supports the use of artificial neural networks to attain high accuracy ratings when classifying misuse trends (Buczak & Guven, 2016). This data set could potentially support such analysis if the timestamp were statistically independent from the other categories. It is possible this could be achieved through limiting the analysis to one sensor’s captured data rather than attempting to incorporate all traffic occurring at all times in the global honeypot network.

**CONCLUSION**

This project used data gathered from low-interaction honeypots to explore the predictability of incoming, hostile network traffic to various destination ports. The high accuracy rate of the resulting Multinomial Naive Bayes model supports the ability to refine company cybersecurity strategies based on information from an economical honeypot solution. Although the model would very likely need to be professionally monitored in a production environment; the risk associated with overfitting, misclassification, or error rate creep is much lower than many other machine learning models currently available. Additionally, it requires fewer computing resources, less training time, and less specialized care in implementation and maintenance than would be required for an artificial neural network. Finally, the predictive responses are easily interpretable and scalable to a large breadth of categories. Ultimately, this model could fit well into a cybersecurity strategy as an economical, preventative measure to monitor and protect a company’s network.

**BIOGRAPHY**

**Samantha Roberts** is a graduate student in the Data Science Program at The George Washington University. Her interests include threat analysis and anomaly detection, social engineering, and behavioral analysis. She has worked in military and government systems in various analytical positions for nine years. She enjoys reading, practicing yoga, and cooking in her spare time.

**Dr. Nima Zahadat** is a professor of data science, information systems security, and digital forensics. His research focus is on studying the Internet of Things, data mining, information visualization, mobile security, security policy management, and memory forensics. He has been teaching since 2001 and has developed and taught over 100 topics. Dr. Zahadat has also been consultant with the federal government agencies, the US Air Force, Navy, Marines, and the Coast Guard. He enjoys teaching, biking, reading, and writing.

**REFERENCES**

Buczak, A. L., & Guven, E. (2016). A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. IEEE Communications Surveys & Tutorials, 18(2), 1153-1176.

Diao, L., Yang, C., & Wang, H. (2012). Training SVM email classifiers using very large imbalanced dataset. Journal of Experimental & Theoretical Artificial Intelligence, 24(2). doi:https://doi.org/10.1080/0952813X.2011.610033

Dua, S., & Du, X. (2016). Introduction. In *Data Mining and Machine Learning in Cybersecurity* (pp. 1-21). Auerbach Publications.

Goseva-Popstojanova, K., Anastasovski, G., & Pantev, R. (2012). Using Multiclass Machine Learning Methods to Classify Malicious Behaviors Aimed at Web Systems. 2012 IEEE 23rd International Symposium on Software Reliability Engineering. doi:10.1109/issre.2012.30

Ip2geo Lookup. (n.d.). Retrieved from <https://ip2geo.org/>

Kaaniche, M., Deswarte, Y., Alata, E., Dacier, M., & Nicomette, V. (2007). Empirical Analysis and Statistical Modeling of Attack Processes Based on Honeypots. ArxIV, 0704(0861).

Nawrocki, M., Wahlisch, M., Schmidt, T., Keil, C., & Schonfelderz, J. (2016). A Survey on Honeypot Software and Data Analysis. ArxIV, 1608(06249), 1st ser.

Prasad, R., & Abraham, A. (2010). Hybrid Framework for Behavioral Prediction of Network Attack Using Honeypot and Dynamic Rule Creation with Different Context for Dynamic Blacklisting. 2010 Second International Conference on Communication Software and Networks. doi:10.1109/iccsn.2010.82

Rapid7. (2018, May 15). Rapid7 Heisenberg Cloud Honeypot cowrie Logs. Retrieved from <https://opendata.rapid7.com/heisenberg.cowrie/>

Roberts, S. (2019). samroberts58/Low-Interaction-Honeypots: Low Interaction Honeypot Traffic (Version v1.0.0). Zenodo.[**http://doi.org/10.5281/zenodo.2248885**](http://doi.org/10.5281/zenodo.2248885)

Sadasivam, G. K., Hota, C., & Anand, B. (2016). Classification of SSH Attacks Using Machine Learning Algorithms. 2016 6th International Conference on IT Convergence and Security (ICITCS). doi:10.1109/icitcs.2016.7740316

Zhan, Z., Xu, M., & Xu, S. (2013). Characterizing Honeypot-Captured Cyber Attacks: Statistical Framework and Case Study. IEEE Transactions on Information Forensics and Security, 8(11), 1775-1789. doi:10.1109/tifs.2013.2279800

**APPENDIX A**

|  |  |  |
| --- | --- | --- |
| **Feature Variable** | **Data Type** | **Description** |
| dst\_port | Object | The targeted destination port |
| ts | Datetime | The timestamp of the record |
| src\_ip | Object | The source IP address associated with the record |
| session | Object | The unique identifier for multiple interactions from the same src\_IP within a specified time period |
| eventid | Object | The transaction request associated with the record |
| sensor | Object | The honeypot identifier |
| Country | Object | The Country registered for the source IP address |
| City | Object | The City registered for the source IP address |