Network Intrusion Detection and Machine Learning

Joel Magee
Department of Computer Science
Hampton University
joel.magee98@gmail.com

Chutima Boonthum-Denecke
Department of Computer Science
Hampton University
chutima.boonthum@hamptonu.edu

Abstract

Overall, this document will serve as an analysis of the combination between machine learning principles and computer network analysis in their ability to detect a network anomaly, such as a network attack. The research provided in this document will highlight the key elements of network analysis and provide an overview of common network analysis techniques. Specifically, this document will highlight a study conducted by the University of Luxembourg and an attempt to recreate the study with a slightly different list of parameters against a different dataset for network anomaly detection using NetFlow data. Alongside network analysis, is the emerging field of machine learning. This document will be investigating common machine learning techniques and implement a support vector machine algorithm to detect anomaly and intrusion within the network. MatLab was an utilized machine learning tool for identifying how to coordinate network analysis data with Support Vector Machines. The resulting graphs represent tests conducted using Support vector machines in a method similar to that of the University of Luxembourg. The difference between the tests is within the metrics used for anomaly detection. The University of Luxembourg utilized the IP addresses and the volume of traffic of a specific NetFlow dataset. The resulting graphs utilize a metric based on the duration of transmitted bytes, and the ratio of the incoming and outgoing bytes during the transmission. The algorithm created and defined metrics proved to not be as efficient as planned against the NetFlow dataset. The use of the conducted tests did not provide a clear classification of an anomaly. However, many other factors contributing to network anomalies were highlighted.

Introduction

In the industry of technology, advancements are frequently manifested. Rapid technological advancements give opportunity for amplified risks and threats. In the past five years, the growth rate of malware infected machines has increased over 45% [3]. The industry of cyber security is developing at a pace to combat the increasing amount of occurrences of exploitations on systems with the intention of "preventing damage to, protection of, and restoration of computers to ensure its availability, integrity, authentication, confidentiality, and nonrepudiation" [8]. National governments have published standards such as the General Data Protection Regulation (GDPR) to provide a clear means of how companies should conduct themselves in relation to cyber-security. Along with the establishment of standards, entities within the cyber-security industry have developed procedures and methods for monitoring and analyzing computer network data for the sake of minimizing exposure to threats and damage upon infiltration. Another emerging field of technology is Machine Learning. Machine learning is defined as an "approach to data analysis that involves building and adapting models, which allow programs to learn through experience"[17]. The following paper will demonstrate the various methods used such as NetFlow and Simple Network Management Protocol (SNMP) for networking monitoring, and investigate the potential impact of machine learning techniques such as Support Vector Machines, upon integration with the intention of anomaly detection.

The need for network analysis has increased due to the threats of cyber-attacks. According to Cisco, some of the most common types of cyber attacks include Malware, Phishing, Man-in-the-middle, and Denial of Service Attacks. Statistics show that DoS attacks are increasing by the year. In the second quarter of 2018, DoS attacks rose 32% from 2017, and increased 46% in 2019 from 2018. A DoS attack acts a flood of data to a central location, in effort to cause the intended victim to malfunction.

Network Analysis

Cisco NetFlow

The Cisco NetFlow network analysis method was developed in 1996, as it was integrated into Cisco's proprietary "Catalyst Operating System Software" for their switches and routers. The Cisco NetFlow network monitoring system is defined as a macro-analytical tool, characterizing large volumes of traffic [21]. The system provides information for each individual Internet Protocol flow that is transmitted through the device, whether a router or a switch. NetFlow captures and records IP packets based on the IP packet attributes; source address, destination address, source port, destination port, layer 3 protocol type, class of service, and interface of hardware. [25]

Cisco Netflow is primarily used as a traffic monitor on one point of the network, therefore does not possess the capability to monitor internal network traffic, nor the capability to develop assessments based on traffic flow. However, coupled with a machine learning algorithm, NetFlow data has the potential to be used as a tool for intrusion detection.

Simple Network Management Protocol

According to RFC 1157, the Simple Network Management Protocol presented as an architectural model, consisting of network management stations and network elements. The concept of a protocol for network management was originally presented in RFC 1052 as a SNMP framework was originally presented in 1988 with RFC 1052, as a recommendation "of the Internet Activities Board (IAB) for the development of network management protocols for use in the TCP/IP environment" [12]. The IAB recommended the proposed SNMP architecture for the sake of aggregating pre-existing management frameworks, such as the Common Management Information Protocol. Also, the IAB wanted to begin researching the workings of inter network management, primarily within internal networks but not excluding commercial networks.

The SNMP protocol was defined in 1989 by RFC 1098. The key elements of SNMP, in relation to management information, were the scope , representation, supported operation, formation and meaning of SNMP exchanges, and the formation of references [26]. Along with aggregating network management information, SNMP adopted the object instances of MIB. The notable object instances associated with SNMP include ifTable , atTable, ipAddrTable, and the ipRoutingTable]. In modern day network management, SNMP is used in a wide range of variances as "the most common class of tools is based on SNMP" [27].

Machine Learning

History of Machine Learning

The first efforts into developing a machine learning algorithm are associated with the 1949 literary works of Donald Hebb. In the book titled The Organization of Behavior, Hebb presents the theory of neuron excitement contributing to the connection of experiences and learning. The theory, developed by Hebb, established the Hebb rule, in which the weight of the connection between two neurons should be managed appropriately in relation to the production of the neurons. In the case that two neurons are able to activate simultaneously, the connection will strengthen [11].

Shortly after the publishing of "The Organization of Behaviour", various other scientific endeavors explored the possibilities of machine learning. In the 1950's, Arthur Samuel of IBM investigated the theory of alpha-beta pruning, using a scoring function associated with the piece positioning of checkers pieces in order to assess the chances of victory. The algorithm later evolved into the minimax algorithm. In 1957, Frank Rosenblatt combined the theories of Arthur Samuel and Donald Hebb to develop "The Perceptron". Although unsuccessful, The Perceptron was developed with the intention to serve as a machine for image recognition. The inception of a model closets to modern day machine learning algorithms was developed in 1967 Marcello Pelillo developed the Nearest Neighbor algorithm. Pelillo utilized the concept of basic pattern recognition, and was used for solving the traveling salesperson's problem [4].

Upon investigation of the nearest neighbor algorithm and the development of neural networks through multi-layering, it is apparent that modern machine learning algorithms have been impacted. The following section will discuss the support vector machine, and Random Forest algorithms.

Machine Learning Methods

Support Vector Machines

The use of the support vector machine algorithm is to identify a "linear separable hyperplane", or a decision boundary

separating members of one class from the other" [16]. The algorithm utilizes support vectors and margins to identify the hyperplane. The primary purpose of SVM's is the classification of data. Data is plotted in an n-dimensional space with a value assigned at each coordinate within the plane. The decision boundary is then used to identify the objective of the function. The plotted data is analyzed to determine the maximum distance to the decision boundary with the minimum distance from the supporting vector. [28]

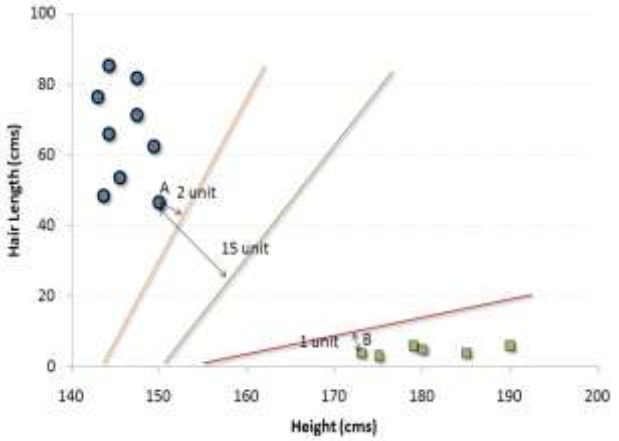

Figure 1. Adapted from *Support Vector Machine - Simplified,* by Tavish Srivastava, 2014, Analytics Vidhya [28]

Random Forest

The Random Forest theory derives from the basis of decision trees. The algorithm is widely used for the purposes of regression and classification. A random forest model consists of smaller trees which develop individualized predictions [23]. The smaller trees are utilized as breaking points to divide data. The process is completed when combining the predictions of the smaller trees to develop a final prediction. A distinction between the algorithms of random forests and simple decision trees, is the aspect of overfitting, of which is not applicable to random forests. A process often referred to as bagging, ensures the voids of overfitting. During the process of bootstrap aggregation, the possible correlations between individual trees is eliminated by sampling a data subset at random [18]. The smaller trees are developed using the algorithm in which the random variables B, are respectively associated with variance σ2, 1/B σ2 and the variables are identically distributed with positive pairwise correlation ρ. ρσ2 +(1 - ρ)/B σ2 [24]

Also, to assist with the removal of correlations, random tree algorithms split the dataset on the premise of randomly specified features within the subset [22]. Upon the addition of a new instance, the features will be assessed by the "mean value of the predictions of all the estimators" to determine the appropriate class for prediction.

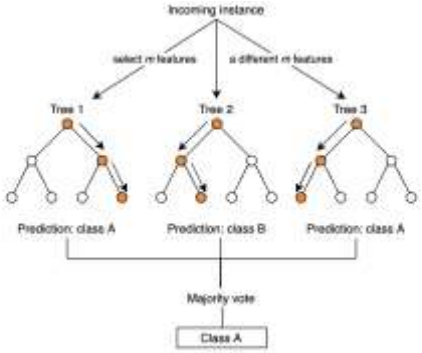

Figure 2. Adapted from Random Forests, by Thomas Wood, DeepAI [22]

NetFlow & Support Vector Machines

As mentioned earlier, NetFlow is the framework created by Cisco systems for network monitoring. NetFlow and Support Vector Machines have yet to be implemented within the industry of cyber-security. Therefore, to assess the effectiveness of Support Vector Machines in relation to NetFlow data, I observed the methods used at the University of Luxembourg, by Cynthia Wagner, Thomas Engel, Jerome Francois and Radu State. The conducted tests were centered around the two parameters of IP addresses, source and destination, and the traffic volume in bytes. It was concluded that the highest success rate of identifying network anomalies was with stealthy DDos, at 93.8 percent. In order to test the data, a unique kernel function was created and associated with One classifying Support Vector Machine.

Method

In order to develop a hands on understanding of the impact of machine learning with network anomaly detection, a basis was adapted on which to utilize the support vector machine algorithm. The support vector machines were combined with the dataset NFS-UNSW-NB15[20]. The dataset consisted of 1,6,23,118 total packet flows. Each packet flow presented the attributes of duration during the session, source IP address, destination IP address, amount of bytes coming into the network and leaving, and the amount of packets that were exchanged during the transmission. In order to effectively train the SVM to appropriately identify a DoS attack, an

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	IPV4_SRC...	L4_SRC...	IPV4_DST...	L4_DST...	PROTO...	L7_PR...	IN_BYT...	OUT_B...	IN_PKTS	OUT_P...	TCP_F...	FLOW_DURATION_MILLIS...	Label	Attack
6	'59.166.0.5'	57349	'149.171.126.1'	53	17	5	146	178	2	2	0	1	0	'Benign'
7	'59.166.0.3'	41560	'149.171.126.7'	80	6	7	690	1272	4	8	25	5	0	'Benign'
8	'59.166.0.5'	29259	'149.171.126.0'	25	6	3	37914	3380	54	42	27	22	0	'Benign'
9	'59.166.0.6'	1813	'149.171.126.1'	53	17	5	130	162	2	2	0	1	0	'Benign'
10	'59.166.0.2'	20139	'149.171.126.7'	80	6	7	690	1272	4	8	25	2	0	'Benign'
11	'59.166.0.6'	54026	'149.171.126.5'	53	17	5	146	178	2	2	0	1	0	'Benign'
12	'59.166.0.6'	48822	'149.171.126.3'	143	6	4	7818	15800	122	128	27	1179	0	'Benign'
13	'59.166.0.6'	1888	'149.171.126.9'	80	6	7	17538	1087890	328	746	27	1139	0	'Benign'
14	'59.166.0.6'	33204	'149.171.126.7'	53	17	5	130	162	2	2	0	0	0	'Benign'
15	'59.166.0.7'	49223	'149.171.126.3'	5190	6	0	1958	2308	22	24	27	7	0	'Benign'
16	'59.166.0.0'	15525	'149.171.126.1'	80	6	7	690	1272	4	8	25	2	0	'Benign'
17	'59.166.0.7'	4507	'149.171.126.7'	22	6	92	3728	5474	32	34	27	6	0	'Benign'
18	'59.166.0.7'	6516	'149.171.126.9'	53	17	5	146	178	2	2	0	1	0	'Benign'
19	'59.166.0.3'	39872	'149.171.126.5'	5190	6	0	1920	4312	22	24	27	7	0	'Benign'
20	'59.166.0.1'	60560	'149.171.126.0'	34044	6	0	8928	320	14	6	27	423	0	'Benign'
21	'59.166.0.5'	20860	'149.171.126.2'	60046	6	0	424	8824	8	12	27	456	0	'Benign'
22	'59.166.0.1'	14055	'149.171.126.1'	26175	17	11	544	304	4	4	0	1	0	'Benign'
23	'59.166.0.1'	12607	'149.171.126.1'	111	17	11	568	304	4	4	0	4	0	'Benign'
24	'59.166.0.1'	34713	'149.171.126.1'	35373	6	11	3936	2456	18	18	27	4	0	'Benign'

Figure 3: Results from SVM and MatLab

emphasis was placed on the relationship between the duration of the flow and the total transmission of bytes exchanged. As the duration of a transmission could be low, the presence of a high amount of bytes exchanged in a short time frame characterizes a DoS attack. 80% of the data set was utilized to train the SVM. The remaining 20% were used for testing purposes. Two axises were calculated to provide an outline for the presentation of data. The dataset would be trained and tested on graphs representing the relationship between the duration of the byte flow and the ratio of incoming and outgoing packets within the dataset. The duration of byte transmission metric was calculated by the following formula ((duration of the flow of bytes)/(total amount of transmitted bytes)). The incoming and outgoing packet ratio metric was calculated by the formula ((incomingt flow of packets)/(outgoing flow of packets)). It was hypothesized that the packet flows that indicated a network anomaly, specifically a Denial of Service attack, would be represented linearly.

To effectively determine the probability of a data point relating to a DoS attack, MatLab was used to create sub-tables from the dataset. The sub-tables were created on the basis of combining the previously mentioned data flow duration - total transmission of bytes relationship, with the classification of the specific data set, such as Benign or DoS. Upon creation of the sub-tables executed was a regression and classification learner application, within MatLab, on the sub-tables.

Results

Figure 3 depicts the ordering and classification of the dataset as the results of training and testing dataset. Figure 4 was captured after the original regression learning without the stream classification set. The yellow markings represent the test data, while the blue markings represent the learned data. The constructed algorithm would have placed DoS attacks higher in the y-axis but lower in the x-axis. The following image depicts the isolated yellow markings of the test data.

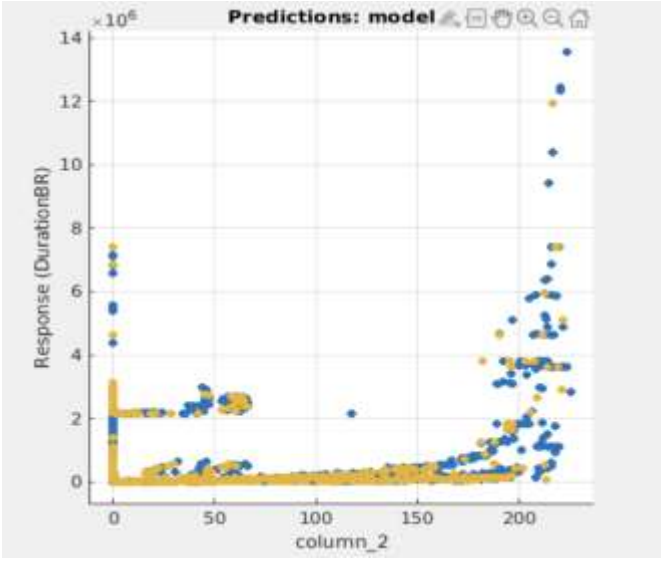

Figure 4: Regression Learning w/o stream classification

Figure 5 depicts the data set used after aggregating the classification data with the theorized byte rate (DurationBR) and input/output packet ratios (IOR). In the graphic, it is apparent that the data followed the similar courses as the assessment prior.

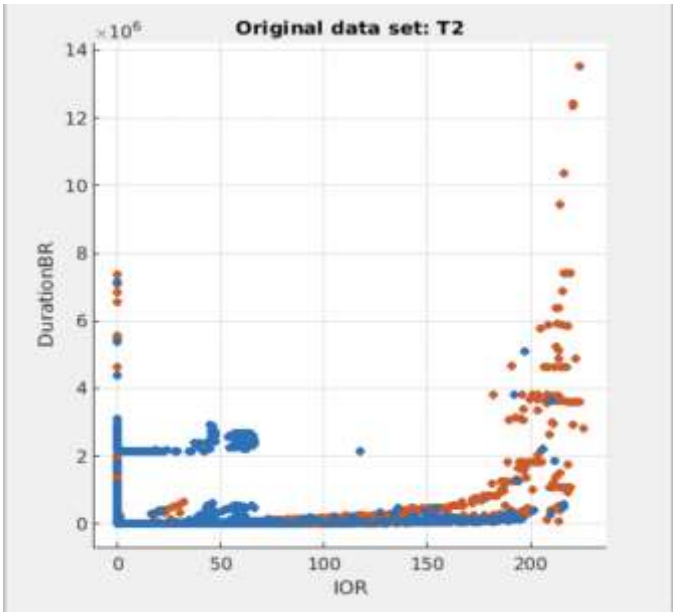

Figure 5: Aggregation Classification Data with DurationRR and IOR

The addition of the classification data allowed for the clearer understanding of the relationship between input/output ratio and duration byte rate in relation to cyber-attacks. For instance, Figure 6, the Denial-of-Service (DoS) attackhad the closest to least input/output byte ratio (IOR), when compared to benign packet streams that has the most IOR.

Conclusion

In conclusion, as demonstrated in related studies and experiments, machine learning algorithms can have a beneficial impact on the analysis and detection of anomalies. However, the appropriate algorithm and selected feature set must be taken into consideration. During investigation outlined in this document, if in the case the algorithm for determining anomalies was modified towards another aspect of the dataset, the machine learning algorithm could likely adapt to the data in a more successful manner. In future works, the selected features on which to base the algorithm must be properly vetted and analyzed to ensure the highest possible level of success during the machine learning process.

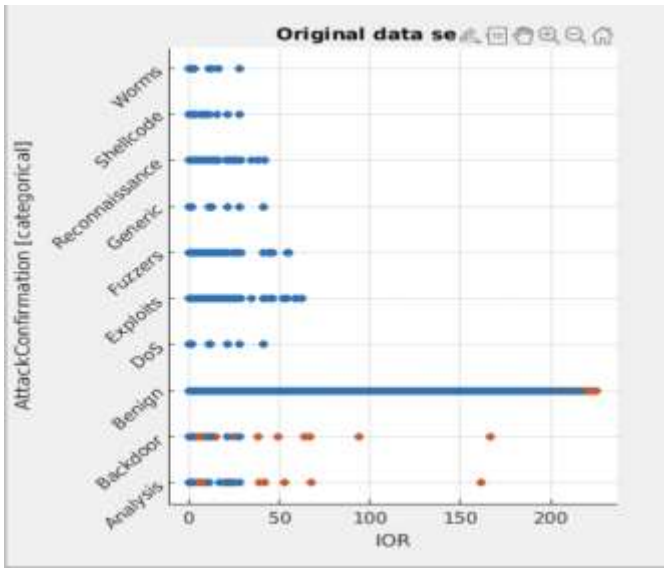

Figure 6: Classification based on Attack Confirmation

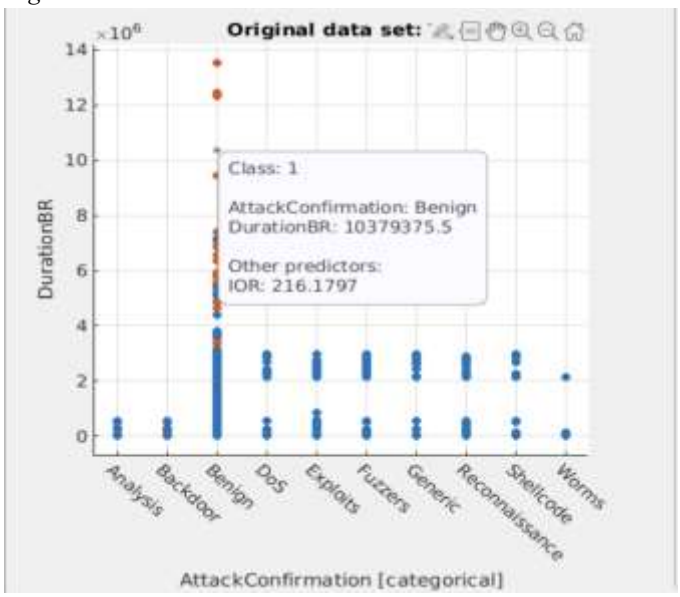

Figure 7: Classification based on DurationBR

Acknowledgements

This work is partly supported by the National Science Foundation CyberCorps: Scholarship for Service program under grant award# 1754054.

References

[1] "10.2 - Support Vector Classifier | STAT 897D." https://online.stat.psu.edu/stat857/node/241/ (accessed Feb. 17, 2021).

[2] R. S. Updated: 2/1/2021, "134 Cybersecurity Statistics and Trends for 2021 | Varonis," *Inside Out Security*, Jan. 13,

2020. https://www.varonis.com/blog/cybersecurity-statistics/ (accessed Feb. 10, 2021).
[3] "2019 Cyber Security Statistics Trends & Data," *PurpleSec*, Nov. 08, 2020. https://purplesec.us/resources/cyber-security-statistics/ (accessed Feb. 03, 2021).
[4] K. D. Foote, "A Brief History of Machine Learning," *DATAVERSITY*, Mar. 26, 2019. https://www.dataversity.net/a-brief-history-of-machine-learning/ (accessed Feb. 17, 2021).
[5] "A Summary of Network Traffic Monitoring and Analysis Techniques." https://www.cse.wustl.edu/~jain/cse567-06/ftp/net_monitoring/index.html#sec2.1.1 (accessed Feb. 02, 2021).
[6] T. Fierro, "Aruba AI Advantage," p. 3.
[7] J. Joyce, "Bayes' Theorem," Jun. 2003, Accessed: Feb. 18, 2021. [Online]. Available: https://plato.stanford.edu/archives/spr2019/entries/bayes-theorem/#1.
[8] C. C. Editor, "cybersecurity - Glossary | CSRC." https://csrc.nist.gov/glossary/term/cybersecurity (accessed Feb. 10, 2021).
[9] S. Williams, "DDoS attacks on the rise despite taking a summer break." https://securitybrief.eu/story/ddos-attacks-on-the-rise-despite-taking-a-summer-break (accessed Feb. 18, 2021).
[10] S. Naseer *et al.*, "Enhanced Network Anomaly Detection Based on Deep Neural Networks," *IEEE Access*, vol. 6, pp. 48231–48246, 2018, doi: 10.1109/ACCESS.2018.2863036.
[11] "Hebbian Learning Rule." http://penta.ufrgs.br/edu/telelab/3/hebbian_.htm (accessed Feb. 17, 2021).
[12] V. G. Cerf, "IAB recommendations for the development of Internet network management standards." https://tools.ietf.org/html/rfc1052 (accessed Feb. 18, 2021).
[13] "Introduction to Cisco IOS NetFlow - A Technical Overview," *Cisco*. https://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/ios-netflow/prod_white_paper0900aecd80406232.html (accessed Feb. 12, 2021).
[14] "IT Management Software - for MSPs and IT Teams," *Kaseya*. https://www.kaseya.com/ (accessed Feb. 11, 2021).
[15] B. Marian *et al.*, "LEMON - LHC Era Monitoring for Large-Scale Infrastructures," *J. Phys.: Conf. Ser.*, vol. 331, no. 5, p. 052025, Dec. 2011, doi: 10.1088/1742-6596/331/5/052025.
[16] "Lesson 10: Support Vector Machines | STAT 897D." https://online.stat.psu.edu/stat857/node/211/ (accessed Feb. 17, 2021).
[17] "Machine Learning," *DeepAI*, May 17, 2019. https://deepai.org/machine-learning-glossary-and-terms/machine-learning (accessed Feb. 11, 2021).
[18] D. S. Elsinghorst, "Machine Learning Basics - Random Forest," *Shirin's playgRound*. /2018/10/ml_basics_rf/ (accessed Feb. 18, 2021).
[19] "MATLAB Online R2020b." https://matlab.mathworks.com/?trial=true&elqsid=1613698232293&potential_use=Student (accessed Feb. 19, 2021).
[20] T. U. of Queensl, A. B. S. Lucia, Q. 4072 +61 7 3365 1111 O. C. U. Ipswich, U. Q. Gatton, U. H. Maps, and D. © 2013 T. U. of Queensl, "ML-Based NIDS Datasets," *School of Information Technology and Electrical Engineering*. https://www.itee.uq.edu.au/research/cyber-security/research-areas (accessed Feb. 18, 2021).
[21] "prod_case_study0900aecd80311fc2.pdf." Accessed: Feb. 12, 2021. [Online]. Available: https://www.cisco.com/c/dam/en/us/products/collateral/ios-nx-os-software/ios-netflow/prod_case_study0900aecd80311fc2.pdf.
[22] "Random Forests," *DeepAI*, Sep. 10, 2020. https://deepai.org/machine-learning-glossary-and-terms/random-forest (accessed Feb. 18, 2021).
[23] "Random forests - classification description." https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm#workings (accessed Feb. 18, 2021).
[24] "randomforest.pdf." Accessed: Feb. 18, 2021. [Online]. Available: https://www.math.mcgill.ca/yyang/resources/doc/randomforest.pdf.
[25] "rstl.1763.pdf." Accessed: Feb. 18, 2021. [Online]. Available: https://royalsocietypublishing.org/doi/pdf/10.1098/rstl.1763.0053.
[26] J. Davin, J. D. Case, M. Fedor, and M. L. Schoffstall, "Simple Network Management Protocol (SNMP)." https://tools.ietf.org/html/rfc1098#page-4 (accessed Feb. 18, 2021).
[27] "SNMP and Beyond: A Survey of Network Performance Monitoring Tools." https://www.cse.wustl.edu/~jain/cse567-06/ftp/net_traffic_monitors2/index.html (accessed Feb. 18, 2021).
[28] "Support Vector Machine | SVM Classification Algorithm," *Analytics Vidhya*, Oct. 03, 2014. https://www.analyticsvidhya.com/blog/2014/10/support-vector-machine-simplified/ (accessed Feb. 17, 2021).
[29] S. Jamshidi, "The Applications of Machine Learning Techniques in Networking," p. 22.