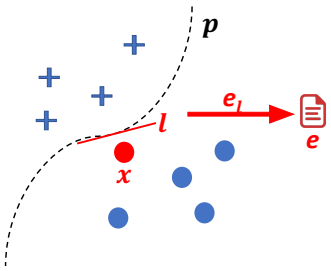

Figure 1: To obtain an explanation e of the outcome of a predictive model p against a given input x, a model l is developed to imitate the local prediction of p around x and an explanation logic e_l is used to reason over l and x.

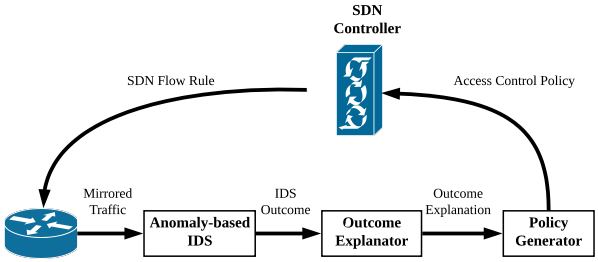

Figure 2: A High-level view of our overall approach that achieves dynamic nework access control.

The contributions of this paper are as follows.

- We investigate the approach to explaining the outcome of anomaly-based IDS through local approximation of decision boundary of the anomaly-based intrusion detection model.
- We demonstrate the feasibility of our approach with a case study through generating a network access control policy based on the outcome explanation of an anomaly-based IDS.

The rest of the paper is organized as follows. In Section 2, we provide the background and describe the details of our technical approach. In Section 3, a case study is presented to demonstrate the feasibility of our proposed approach. we conclude and discuss our future work in Section 4.

2 METHODOLOGY

2.1 Background

Formally, a predictive model for anomaly-based IDS is a function $p : X^{(m)} \rightarrow Y$, which takes an instance $x \in X^m$ with m features as input and outputs a score $y \in Y$ indicating the deviation between the input instance and a normal instance. Conventionally, $p(x) = y$ denotes that y is an outcome of the prediction of p corresponding to the input x. The task of explaining an outcome of a predictive model can be defined as follows. Given an input x and a predictive model p, we need to find an interpretable model l that imitates the prediction of p locally against x. Then, an explanation e is obtained through some explanation logic e_l reasoning over x and l. Figure 1 illustrates each concept in the aforementioned process.

There are two key components in the above process: the interpretable model and the explanation logic. There is a set of models, such as *decision tree* [12, 16] and *linear models* [14, 17], believed to be easily understandable and interpretable for humans. For explanation logic, there exist three major techniques: *saliency map* [22, 24], *decision rules* [18], and *feature importance* [14, 17].

In this work, we employ linear models to approximate the local decision boundary of the original predictive and obtain the explanation through feature importance.

2.2 Challenge and Technical Overview

Despite SDN introduces great programmability to the network, it is still limited in considering network anomalies when controlling the network. This limitation is caused fundamentally by the fact that the SDN controller is not designed to analyze the network traffic in depth like IDS.

To overcome this shortcoming, an anomaly-based IDS is introduced to operate on the data plane to directly analyze the traffic in depth and report any network anomalies to aid the SDN controller controlling the network. Deriving access control policies from the results of a signature-based IDS [1, 2] could be straight forward since the signatures are working in an *if-match-then-action* manner [19], which can be easily mapped to an access control policy. However, there exist several limitations of signature-based IDS, including being unable to detect zero-day security threats and suffering from signature base explosion.

In the following, we achieve dynamic network access control through the combination of SDN and anomaly-based IDS. Figure 2 depicts the high-level idea of our overall approach. We mirror a copy of network traffic from the SDN switch to an *anomaly-based IDS*. If the outcome of the anomaly-based IDS indicates an anomaly, it is sent to the *outcome explanator* to get explained. The explanation of the outcome is then processed by the *policy generator* to generate the access control policy. The generated access control policy can be sent to the SDN controller via a network control framework [8]. The SDN controller finally installs SDN flow rules into the SDN switch according to the access control policy.

2.3 Technical Details

Existing work has already demonstrated how to develop anomaly-based IDS using Deep Neural Network (DNN) [20, 21, 23], and a network control framework has been presented in [8] to allow IDS communicate with the SDN controller. Therefore, our focus is the outcome explanation and policy generation.

Anomaly-based IDS Outcome Explanation. As is shown in Figure 1, outcome explanation consists of two major steps: *i)* training a model to approximate the local decision boundary of the target predictive model; and *ii)* reasoning on the trained model and the given input based on some explanation logic.

To approximate the local decision boundary of the target predictive model, we first synthesize a set of data samples around x as described in LEMNA [14] and utilize the target predictive model to predict the label for each synthetic data sample. Then we can use those synthetic data samples to train a linear regression model l defined as:

$$l(\mathbf{x}) = \boldsymbol{\alpha}\mathbf{x} + \epsilon \tag{1}$$

Feature	dur.	proto.	service	flag	...	dsthost_serror_rate*	dsthost_srv_serror_rate*	dsthost_rerror_rate*	dsthost_srv_rerror_rate*
Record1	0	tcp	private	S0	...	1	1	0	0
Record2	0	tcp	imap4	REJ	...	0	0	1	1

Table 2: Explanation for Neptune SYN-FLOOD attack. The outcome explanation marks the most important feature as red, followed by orange, yellow, lime, cyan. The top-4 features are marked with*.

(dst_host_serror_rate) and *(dst_host_srv_serror_rate)* features represent the percentage of connections that have SYN errors. The *(dst_host_rerror_rate)* and *(dst_host_srv_rerror_rate)* features represent the percentage of connections that have REJECT errors. In reality, *Neptune* is a typical *SYN FLOOD* attack, which is usually fingerprinted by both SYN and REJECT errors. This knowledge is identical to our explanation, thus validates that our anomaly-based IDS is trustworthy and the explanation is correct.

3.3 Policy Generation

From the above explanation we can see that the key features of those abnormal records are highly related to the *SYN* field. To prevent this type of attacks we can generate a policy with deny action for those instances with *SYN* features. Each record in the *NSL-KDD* dataset represents one connection. Each connection can be identified by the source and destination IPs, ports and protocol types [1]. Below is an access control policy generated according to our explanation that drops the network packets related to the *Neptune* attack.

<filters=(src_ip=192.168.1.2, dst_ip=192.168.1.3, ip_proto=6, tcp_flags=0x02), actions=(drop)>

The source IP, destination IP, and the protocol type can be derived from the abnormal connection record. Worth noting, the *tcp_flags* field in the *filters* is set to 0x02 (*SYN* flag set) because the features included in the explanation are all related to *SYN*. This policy will be sent to the SDN controller, which then generates and installs corresponding SDN flow rules into the SDN switch to achieve dynamic network access control.

4 CONCLUSION AND FUTURE WORK

This paper introduces a method to explain the outcome of anomaly-based IDS via an interpretable model and generate the network access control policies based on the explanation. In our case study, we show that the proposed method produces trustworthy outcome and is useful to generate policies for dynamic network access control.

In our future work, we will investigate better explanation approaches to handle the dependency among multiple records to improve our explanation accuracy. In addition, the policy generation process should be formalized to enable full automation. We will also test on real-world network traffic considering different attack types.

ACKNOWLEDGMENTS

This work was partially supported by grants from National Science Foundation (NSF-OAC-1642143, NSF-CNS-1700499, and NSF-DGE-1723663).

[1] *NSL-KDD* dataset does not include the IPs in each record, but in practice the IDS can easily capture the IP information and embed it into the outcome.

REFERENCES

[1] 2018. Snort Network Intrusion Detection & Prevention System. https://snort.org/.
[2] 2018. Suricata IDS. https://suricata-ids.org/.
[3] 2019. KDD Cup 1999 Data. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.
[4] 2019. Keras: The Python Deep Learning library. https://keras.io/.
[5] 2019. NSL-KDD dataset. https://www.unb.ca/cic/datasets/nsl.html.
[6] 2019. TensorFlow. https://www.tensorflow.org/.
[7] Amina Adadi and Mohammed Berrada. 2018. Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI). *IEEE Access* 6 (2018), 52138–52160.
[8] Johanna Amann and Robin Sommer. 2015. Providing Dynamic Control to Passive Network Security Monitoring. In *Proceedings of the 18th International Symposium on Research in Attacks, Intrusions, and Defenses-Volume 9404*. Springer-Verlag New York, Inc., 133–152.
[9] Rocky KC Chang. 2002. Defending against flooding-based distributed denial-of-service attacks: a tutorial. *IEEE communications magazine* 40, 10 (2002), 42–51.
[10] Zhiyong Cheng, Xiaojun Chang, Lei Zhu, Rose C Kanjirathinkal, and Mohan Kankanhalli. 2019. MMALFM: Explainable recommendation by leveraging reviews and images. *ACM Transactions on Information Systems (TOIS)* 37, 2 (2019), 16.
[11] Juan Deng and Hongda Li. 2017. On the Safety and Efficiency of Virtual Firewall Elasticity Control. In *24th Network and Distributed System Security Symposium (NDSS 2017)*.
[12] Alex A Freitas. 2014. Comprehensible classification models: a position paper. *ACM SIGKDD explorations newsletter* 15, 1 (2014), 1–10.
[13] Timon Gehr, Matthew Mirman, Dana Drachsler-Cohen, Petar Tsankov, Swarat Chaudhuri, and Martin Vechev. 2018. Ai 2: Safety and robustness certification of neural networks with abstract interpretation. In *Security and Privacy (SP), 2018 IEEE Symposium on*.
[14] Wenbo Guo, Dongliang Mu, Jun Xu, Purui Su, Gang Wang, and Xinyu Xing. 2018. Lemna: Explaining deep learning based security applications. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 364–379.
[15] Hongxin Hu, Wonkyu Han, Gail-Joon Ahn, and Ziming Zhao. 2014. FLOWGUARD: building robust firewalls for software-defined networks. In *Proceedings of the third workshop on Hot topics in software defined networking*. ACM, 97–102.
[16] Johan Huysmans, Karel Dejaeger, Christophe Mues, Jan Vanthienen, and Bart Baesens. 2011. An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decision Support Systems* 51, 1 (2011), 141–154.
[17] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 1135–1144.
[18] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-precision model-agnostic explanations. In *AAAI Conference on Artificial Intelligence*.
[19] Martin Roesch and Chris Green. 2016. Snort Users Manual 2.9. 8.2.
[20] Nathan Shone, Tran Nguyen Ngoc, Vu Dinh Phai, and Qi Shi. 2018. A deep learning approach to network intrusion detection. *IEEE Transactions on Emerging Topics in Computational Intelligence* 2, 1 (2018), 41–50.
[21] Tuan A Tang, Lotfi Mhamdi, Des McLernon, Syed Ali Raza Zaidi, and Mounir Ghogho. 2018. Deep recurrent neural network for intrusion detection in sdn-based networks. In *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*. IEEE, 202–206.
[22] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*. 2048–2057.
[23] Chuanlong Yin, Yuefei Zhu, Jinlong Fei, and Xinzheng He. 2017. A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access* 5 (2017), 21954–21961.
[24] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2921–2929.