A Dynamic Machine Learning Scheme for Reliable Network-based Intrusion Detection

Eduardo K. Viegas, Everton de Matos, Paulo R. de Oliveira, and Altair O. Santin

Abstract Several works have proposed highly accurate machine learning (ML) techniques for network-based intrusion detection over the past years. However, despite the promising results, proposed schemes must address the high variability of network traffic and need more reliability when facing new network traffic behavior. This paper proposes a new dynamic and reliable network-based intrusion detection model implemented in two phases. First, the behavior of to-be-classified events is assessed through an outlier detection scheme to reject potentially new network traffic, thus, keeping the system reliable as time passes. Second, classification is performed through a dynamic selection of classifier to address the high variability of network traffic. Experiments performed in a new dataset composed of over 60 GB of network traffic have shown that our proposed scheme can improve detection accuracy by up to 33% when compared with traditional approaches.

1 Introduction

Over the last few years, the number of network-based attacks significantly increased and is still on the rise. As an example, according to a Kaspersky report [11], the occurrence of distributed denial-of-service attacks has rose by 48% in the third quarter

Eduardo K. Viegas
Secure Systems Research Center, Technology Innovation Institute (TII), United Arab Emirates, Abu Dhabi, e-mail: eduardo@ssrc.tii.ae

Everton de Matos
Secure Systems Research Center, Technology Innovation Institute (TII), United Arab Emirates, Abu Dhabi, e-mail: everton@ssrc.tii.ae

Paulo R. de Oliveira
Pontifícia Universidade Católica do Paraná, PUC-PR e-mail: paulo.oliveira@ppgia.pucpr.br

Altair O. Santin
Pontifícia Universidade Católica do Paraná, PUC-PR e-mail: santin@ppgia.pucpr.br

of 2022, showing that current security mechanisms have been unable to adequately secure systems. In general, network operators resort to network-based intrusion detection systems (NIDS) to detect this growing number of threats, typically implemented through either two approaches [12]. On the one hand, *misuse-based* techniques detect attacks based on a previously defined set of well-known attack patterns; hence, it can only detect previously known threats, leaving systems insecure against new kinds of attacks. On the other hand, *behavior-based* approaches signals misconducts according deviations of the normal expected behavior, thus, it is able to detect new attacks as long as they behave significantly differ from the normal behaviors.

As a consequence, due to the ever increasing number of newly discovered attacks, significant research efforts have been conducted on *behavior-based* intrusion detection, wherein authors typically resort to machine learning (ML) techniques implemented as a pattern recognition task. To achieve such a goal, proposed detection mechanisms are implemented through a three-phase process, namely model *training*, *validation*, and *testing* [18]. The model *training* goal is to extract a behavioral ML model from a training dataset, whereas the *validation* phase allows researchers to fine-tune their scheme, such as performing feature selection and hyperparameter optimization. Finally, the model accuracy performance is estimated through a *test* dataset, which is then assumed to be experienced when it is deployed on production environments.

Unfortunately, networked environments presents a plethora of challenges when compared to areas that ML have been successfully applied. The behavior of network traffic is highly variable, while also changes over time, a situation that can be caused either due to the discovery of new attacks or even due to the provision of new services [5]. The non-stationary behavior of network traffic presents a significant challenge to the reliability of designed ML-based intrusion detection schemes. On the one hand, due to the high variability of network traffic behavior, the building of a realistic training dataset becomes an unfeasible task. On the other hand, even if the model is able to generalize the behavior of the network traffic, due to the its non-stationary behavior, hard to-be-conducted ML model updates will need to be periodically executed.

Therefore, ensuring that designed ML-based intrusion detection schemes are able to generalize the behavior of network traffic, while also withstanding long periods of time without frequent model updates is a must. In contrast, current proposed techniques usually pursues the highest accuracy on a given dataset, in general, incurring on significant tradeofs on model generalization capabilities [3]. As a result, ML-based intrusion detection techniques, despite the promising reported results, remains mostly as a research topic, rarely being deployed in real-world applications.

In light of this, this paper proposes a new dynamic and reliable network-based intrusion detection model implemented in two stages. First, the classification task is performed through a dynamic selection of classifier scheme, Thus, only the most suitable subset of classifiers are used for the labeling task. Second, unreliable classifications are properly identified through a novelty detection module, which signals unknown network event behaviors that should be used for model retraining pur-

poses. As a consequence, our proposed model is able to address the highly variable nature of network traffic, through the dynamic selection of classifier, while also adequately identify new network behavior through our novelty detection, maintaining the system accurate and reliable.

The main contributions of this paper are as follows:

- An evaluation of the detection reliability of traditional ML-based techniques in face of variable and new network behavior. The experiments shows that current approaches are unreliable for intrusion detection task in real-world network conditions.
- A new dynamic and reliable network-based intrusion detection model able to withstand and identify unknown network traffic behavior.

The remainder of this paper is organized as follows. Section 2 introduces the fundamentals on ML-based intrusion detection and its challenges, while Section 3 provides the related works. Section 4 evaluates the reliability of current ML-based techniques. Section5 describes our proposed model, while Section 6 evaluates it. Section 7 concludes our work.

2 Preliminaries

A typical network-based intrusion detection system (NIDS) is composed by four sequential modules [12]. The first module namely *Data Acquisition* collects network events that will be subsequently evaluated. The behavior of the collected data is extracted by a *Feature Extraction* module. In general, network traffic behavior is depicted through network flow features, which summarizes the communication between network entities in a given time window. For instance, the number of exchanged network packets between two hosts over the last 2 seconds. The built feature vector is used as input by a *Classification* module, which adequately label the network flow as either *normal* or *intrusion*, e.g. by applying a ML model. Finally, *intrusion*-classified events are signaled to the network operator by the *Alert* module.

Several approaches can be used for the classification task in NIDS, wherein the most promising approach resort to machine learning (ML) techniques, as a pattern recognition task [17, 6]. In such a case, the network operator must collect huge amounts of network traffic, and adequately label them as either *normal* or *attack* for the model building task. However, event labeling of network traffic is not easily achievable, and typically demands human-assistance, which is not always available [16]. Therefore, building a realistic training dataset for ML-based NIDS demands significant network operator effort and time.

Notwithstanding, the behavior of network traffic is highly variable while also changes significantly as time passes, a situation caused either by the discovery of new attacks, or even the provision of new services. As a result, the built ML model becomes outdated as time passes, increasing its error rates and affecting its reliability to the network operator [10]. To address such a challenge, the ML model must

be periodically updated, which often demands several days or even weeks to be conducted, due to the challenges related to the training dataset building task. Therefore, although widely used in several fields, such as image classification, fraud detection, and medical diagnosis, ML is rarely used in production for network-based intrusion detection, remaining mostly as a research topic.

3 Related Works

Over the last years, a plethora of works have proposed highly accurate ML-based techniques for network traffic classification tasks [12, 9]. In general, authors focus their efforts on achieving the highest accuracy on a given intrusion dataset, often neglecting how that reflects under real-world conditions. For instance, F. Salo *et al.* [15] make use of an ensemble of classifiers coped with a feature reduction technique to improve the system's accuracy on a variety of intrusion detection datasets. The authors were able to improve accuracy when compared to related approaches, however, no research effort was given on the variability and changes on network traffic. Another ensemble-based approach was proposed by J. Gu *et al.* [7] that makes use of support vector machine (SVM) coped with feature augmentation. Similarly, their scheme is able to improve accuracy while overlooking the applicability of their techniques. N. Moustafa *et al.* [13] makes use of an ensemble of three widely used ML classifiers to improve accuracy in network traffic classification. The authors shows that ensemble-based techniques can improve accuracy, however, they do not address the network traffic variability neither how new network traffic behavior affects their scheme.

In general, to address environments with non-stationary behavior, related works resort to concept drift detection mechanisms. G. Andresini *et al.* [2] proposes a concept-drift-based technique to detect new network traffic behavior in a well-known intrusion dataset. Their approach is able to signal new network traffic that can affect the reliability of deployed intrusion detection techniques. Unfortunately, the authors assume a supervised setting, wherein the network operator can provide the event label as needed. O. Abdel Wahab [21] identifies concept drift making use of a feature reduction technique coped with a variance detection scheme. The author is able to identify new network traffic behavior in a unsupervised setting, however, no action is taken over the newly network traffic behavior. G. Andresini *et al.* [1] proposed a concept drift detection mechanism to identify new network traffic behavior. Their proposed model makes use of a deep neural architecture integrated with a concept drift detection approach to signal new network traffic. Unfortunately, the authors assume a supervised setting for identification of new traffic.

The high variability of network traffic behavior is rarely addressed by related works. Z. Chkirbene *et al.* [4] proposes a dynamic intrusion detection scheme through feature selection. The authors make use of a feature ranking approach to build new ML models over time, while discarding outdated ones as time passes. Although their approach works in unsupervised setting, the authors neglect how

Table 1: Testbed behavior variations, over time, according to each considered scenario. Both normal and attacker behaviors vary as time passes.

Scenario	Time Window (minutes)	Attacker Behavior	Normal Behavior
Attack (Serv. Scan)	*zero* to 5	*OS* and *service fingerprint*	100 benign clients performs periodic queries on HTTP, and SNMP services
Attack (Portscan)	5 to 10	*udpscan*, *synscan*, *nullscan*, *finscan*, *xmasscan*, and *ackscan*	
Attack (Vuln. Scan)	10 to 15	*vulnerability scan*	
Normal (Content)	15 to 20	*OS* and *service fingerprint*	Requested service content differs from previous scenario
Normal (Service)	20 to 25		100 benign clients performs periodic queries on SMTP, NTP and SSH services

network traffic behavior changes affects their technique, as they make use of an outdated dataset for evaluation purposes. R. Heartfield *et al.* [8] proposed a self-configurable intrusion detection scheme based on reinforcement learning. The author's approach makes use of a reinforcement learning technique to continuously adjust the underlying model's parameters to address new network traffic behavior. Unfortunately, their proposal assume a supervised setting to identify new network traffic. F Pinage *et al.* [14] proposed a dynamic classifier selection for drift detection that can be applied for intrusion detection. The author's approach uses a semi-supervised drift detector that makes use of an ensemble of classifiers. However, the authors do not evaluate their technique on a realistic intrusion dataset.

As a result, although widely studied in the literature, ML-based intrusion detection remains mostly as a research topic, despite the promising results reported by the related works. This is because, in general, authors neglect how their proposed schemes can be used in real-world conditions, wherein the network traffic behavior is highly variable while also changes as time passes.

4 Problem Statement

In this section we investigate how widely used ML-based intrusion detection techniques perform when being used under real-world network conditions. More specifically, we first introduce our used dataset that presents real-world network characteristics, then, we evaluate the detection performance when ML-based techniques are used for intrusion detection over it.

The next subsections present our dataset and our evaluation.

4.1 A Fine-grained Intrusion Detection Dataset

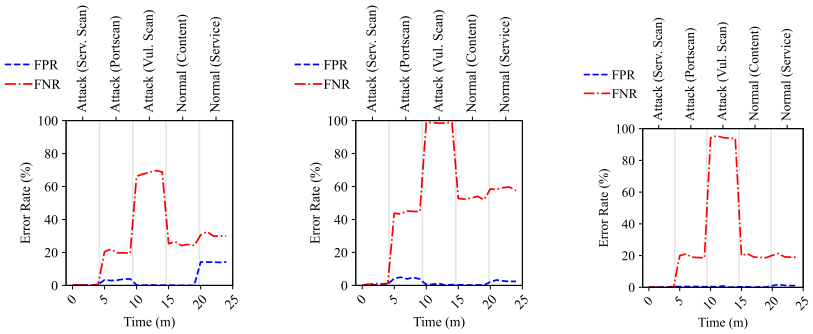
Realistic intrusion detection datasets are a must in order to ensure that proposed schemes are adequately built and evaluated [12]. Unfortunately, making use of outdated datasets with known flaws have become a common practice in the literature [19]. Yet, apart from making use of a realistic dataset, with real and valid network traffic behavior, researchers must ensure that they consider the non-stationary behavior of network traffic. Thus, to adequately evaluate ML-based intrusion detection approaches, our work make use of the *Fine-grained Intrusion Dataset* (FGD), which was built using the methodology presented in [20]. The FGD is made of a variety of network flows that usually is evidenced in real network environments, wherein designed ML models must be able to address. As a result, designed techniques can be adequately assessed considering the network traffic variability of production environments.

To achieve such a goal, the FGD dataset was built making use of a controlled environment with 100 client machines, which are responsible to generate the normal network traffic according to a number of application protocols, as well as several attackers that generates the malicious network traffic using standardized pentesting tools [20]. Each machine executed the service requests and the attacks on a honeypot server used to generate adequate response and also collect the generated network traffic. Table 1 shows the overview of the testbed behavior variation as time passes, concerning the normal and attack network traffic. A detailed description of each network profile can be found in [20]. At total, the used testbed is composed by ≈ 60 GB of network traffic, account for a total of 5 different network traffic behavior.

4.2 Chasing a Moving Target

In this section, we evaluate how changes in the network traffic behavior can affect ML-based intrusion detection schemes. To achieve such a goal, we take into account a training setting composed of network-related probing attacks, such as service scanning (Table 1, *Attack (Serv. Scan)*). To build the selected ML classifiers, we use as input the data that occurs on the first dataset minute. Therefore, the classifiers are built considering application-level network probing attacks, such as OS and service scanning. Each network flow is represented by a feature vector composed by 50 features [20]. Each feature summarizes the network data exchanged between a two given hosts in our testbed and their related services considering a 2-second time window.

We select three widely used ML classifiers for network-based intrusion detection task, namely *Bagging*, *Decision Tree*, and *Random Forest*. The Bagging and Random Forest classifiers were implemented making use of 100 decision trees as their base-learners, where each base learner makes use of *gini* as node split quality metric. The Decision Tree classifier was implemented using *gini* as node split quality metric. A random undersampling without replacement is used at the train-



(a) Bagging (b) Decision Tree (c) Random Forest

Fig. 1: The accuracy behavior, over time, on the evaluated testbed of widely used ML classifiers. Classifiers are trained using only *Attack (Serv. Scan)* data occurred at time window ranging from *zero* to 1 minute.

ing procedure to balance the occurrence between the classes. The classifiers were implemented through $scikit-learn$ API v0.24.

The classifiers are evaluated concerning their false-positive (FPR) and false-negative (FNR) rates, where the FP denotes the ratio of normal events misclassified as an attack. In contrast, the FN represents the ratio of attack events misclassified as normal ones.

Figure 1 shows the error rate variation as time passes for the selected classifiers on our dataset. It can be noted that the selected techniques were able to provide significantly low error rates when facing behaviors already present during the training phase (Fig. 1, 0^{th} to 5^{th} minute). In practice, the selected approaches presented an average of 0.2% and 0.1% of FPR and FNR rates respectively. Unfortunately, over time as the behavior of the underlying network traffic also evolves, the selected approaches becomes unable to cope the intrusion detection. In practice, the selected techniques were only able to present low error rates (false rates $< 5\%$) when being subject to the same kinds of network traffic behavior they were trained with.y

Consequently, evaluated approaches were not able to deal with new kinds of network traffic behavior, demanding model updates to be frequently conducted. Such a situation is caused by changes in the network traffic, either caused by new services or by new attacks beig generated, and as a result, increasing the ratio of false alarms the system produces as time passes.

5 A Dynamic and Reliable Network-based Intrusion Detection Model

In light of this, to address the non-stationary and highly variable nature of real-world networks, we propose a dynamic and reliable network-based intrusion de-

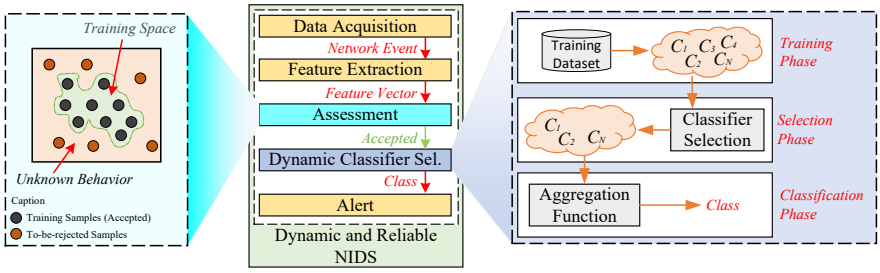


Fig. 2: Proposed dynamic and reliable network-based intrusion detection system model.

tection model. The overview of our proposed scheme is shown on Figure 2 and is implemented in two phases.

First, the highly variable nature of network environments is addressed through a dynamic selection of classifier approach. As a consequence, our proposed scheme is able to proactively select the most suitable subset of classifiers that should be used for classification purposes, based on the to-be-classified event behavior. The main insight of making use of dynamic selection of classifier is to enable our system to adapt according to the network event behavior, which is usually highly variable in production environments. Second, to address new network traffic behavior as time passes, our proposed scheme performs the classification assessment of to-be-classified events. The classification assessment goal is to ensure that our system is only used to classify events that are similar to those used during the training phase, thus, ensuring that only reliable classifications are performed by our model. Consequently, our proposed model is able to address the highly variable nature of network environments, while also keeping its reliability in face of new network traffic behavior.

The next subsections further describe our proposed model including the modules that implements it.

5.1 Classification Assessment

The behavior of real-world network traffic is highly variable while also changes as time passes. As a result, the building of a reliable ML-based technique for intrusion detection becomes a challenging and unfeasible task, demanding that designed techniques are able to cope with unseen network traffic. Unfortunately, ML-based techniques performs a decision for every given input, regardless if it can be reliably made by the underlying ML model, leaving ML-based intrusion detection techniques unreliable in such situations.

To address such a shortcomming, our proposed scheme makes use of a *Assessment* module, as shown in Figure 2. The module goal is to ensure that our system only performs the classification task on events behaves similarly to those used during our system training phase. Our main assumption is that, in order to keep the system reliability, only events that are known to the underlying ML model should be used for classification task. To achieve such a goal, our scheme builds an outlier detector to signal events that are not similar to the events used for training purposes. The outlier detector is build upon the training dataset, hence, it flags as an anomaly events that are significantly different from the training dataset. Signaled events are not used for classification purposes, and, as a consequence, are rejected by our system in order to keep its reliability.

Therefore, our model is able to identify new network traffic behavior in an unsupervised fashion, as events that are identified as anomalies are rejected by our system. As a main contribution our scheme is able to identify the network events that should be evaluated by the network operator, e.g. to update the ML model.

5.2 Classification Pipeline

Reliable network-based intrusion detection schemes must ensure their generalization in face of the highly variable network traffic in production environments. However, widely used ML-based techniques are static after the training phase, in practice, they are not able to select which model should be used for classification based on the newly input data. To address such a challenge, our proposed scheme performs the classification task through a dynamic selection of classifier approach. As a result, only the most suited subset of classifiers are used to perform the event labeling task, hence, adjusting our classification procedure according to the network traffic behavior.

Our proposed model classification pipeline is shown in Figure 2, and it starts with a to-be-classified network event collected by a *Data Acquisition* module. The collected event behavior is extracted by a *Feature Extraction* module which compounds a related feature vector to be used by our system. The built feature vector is first evaluated by our *Assessment* module, which reject outliers based on the used training data (see Section 5.1), while inliers are forwarded for the classification task. Accepted events are used as input by our dynamic classifier selection module.

Our classification procedure considers a traditional dynamic selection of classifiers as shown in Figure 2. Therefore, a pool of classifiers (C) of size N is built over a training dataset. At classification phase (Fig. 2, *Selection Phase*), a subset of classifiers are selected to be used for the classification task according to the input event, e.g. through a competence region map. The output of the selected subset is used by an aggregation function, which outputs a corresponding event class (Fig. 2, *Classification Phase*). Finally, intrusion-detected events are signaled to the operator by an *Alert* module.

5.3 Discussion

Our proposed model aim to address the challenges of real-world networked environments in which the network behavior is highly variable while also changes over time. On the one hand, to address the highly variable nature of network behavior, our proposed scheme performs the classification task through a dynamic selection of classifier approach, selecting the most suited subset of classifiers for the classification of each event. On the other hand, new network traffic behavior is identified by our proposed assessment approach, which rejects new network traffic behavior through an outlier detection scheme. As a result, our proposed model is able to address the non-stationary behavior of network traffic, while ensuring the systems's reliability as time passes.

6 Evaluation

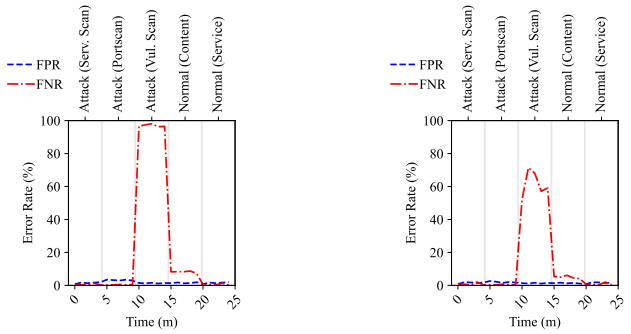
In this section we further investigate the performance of our proposed model when compared to traditional ML-based intrusion detection schemes. More specifically, our evaluation aims at answering the following research questions (RQ):

- ***(RQ1)*** *What is the classification performance of our proposed dynamic selection of classifier approach?*
- ***(RQ2)*** *How does our proposed model perform with our assessment technique?*
- ***(RQ3)*** *How does our proposed model perform when compared to traditional approaches?*

The next subsections further describes our model building procedure and the performed evaluations.

6.1 Model Building

Our proposed model was built and evaluated making use of the same settings evaluated previously (see Section 4.2). More specifically, we build our scheme using the FGD (Table 1) dataset, and evalute its performance while changing the scenario behavior. To achieve such a goal, our *Assessment* module (Fig. 2, *Assessment*) was implemented making use of the Isolation Forest outlier detector. The Isolation Forest was implemented with 100 base estimators, and 256 samples per estimator. The outlier detector was trained using the whole training dataset, i.e., first minute of the FGD dataset, and implemented through the $scikit-learn$ API v0.24. The dynamic selection of classifier was implemented with the *k-Nearest Oracle Union* (KNORA-U) method, with a Bagging classifier as base pool, and 7 neighbors for the competence region computation. The KNORA-U was implemented through the *deslib* API *v*.0.3.



(a) Without Assessment (b) With Assessment

Fig. 3: The accuracy behavior, over time, on the evaluated testbed of our proposed scheme. Our approach is trained using only *Attack (Serv. Scan)* data occurred at time window ranging from *zero* to 1 minute.

6.2 A Reliable and Dynamic NIDS

Our first experiment aims at answering RQ1, and evaluates our proposed model accuracy performance on FGD dataset while not making use of our assessment technique. More specifically, we first evaluate how our dynamic selection of classifier technique performs when being subject to our dataset behavior variations. Figure 3a shows the classification accuracy of our model without assessment. It is possible to note that our dynamic selection of classifier technique is able to significantly decrease the intrusion detection error rate when compared to traditional techniques (Fig. 3a *vs*. 1). More specifically, our scheme presented a high error rate only on one scenario (*Attack (Vul. Scan*), while the traditional approaches significantly decrease their accuracy as soon as a new behavior is evidenced. Therefore, the dynamic selection of classifier can be used to address the network traffic behavior variability.

Our second experiment aims at answering RQ2, and evaluates our proposed model with our assessment technique. In such a case, we only evaluate the events that are deemed as inlier by our assessment model (Fig. 2, *Assessment*). Figure 3b shows the classification accuracy of our model while rejecting outlier events identified by our assessment technique. It can be noted that our assessment approach is able to decrease further the error rate on the *Attack (Vul. Scan* scenario by up to 35%. As a consequence, the assessment approach can be used to improve the system's reliability when a new network traffic behavior is experienced. In addition, the network operator can use the assessment technique to trigger the model update, addressing the increased error rate when the system is deployed in production.

Finally, to answer RQ3, we compare the accuracy performance of our proposed scheme versus the traditional approaches. Table 2 shows the error rate performance of the selected techniques. Our proposed model significantly improve the accuracy when compared to other approaches. For instance, at the most difficult scenario

Table 2: Accuracy of evaluated techniques according to the testbed behavior

.

Testbed Behavior	**Detection Measure**	***Detection Approach Error Rate (%)***			
		Bagging	**Decision Tree**	**Random Forest**	**Proposed Approach**
Attack (Serv. Scan)	FPR	0.8	0.3	0.4	0.0
	FNR	1.2	1.3	0.3	0.4
Attack (Portscan)	FPR	3.8	4.5	1.2	2.3
	FNR	22.2	45.3	23.1	0.4
Attack (Vuln. Scan)	FPR	0.3	0.4	0.7	0.3
	FNR	68.2	97.3	92.1	64.2
Normal (Content)	FPR	0.3	0.1	0.2	0.4
	FNR	25.2	53.2	22.1	4.5
Normal (Service)	FPR	18.3	3.4	2.1	1.3
	FNR	28.2	58.2	21.2	3.2

(*Attack (Vul. Scan*) our scheme was able to decrease the error rate by up to 33.1%, while also presenting a significantly low error rate on the other scenarios. Therefore, our proposed model was able to address the high variability of network traffic, while also being able to identify new network traffic behavior that should be used for model updates.

7 Conclusion

Network-based intrusion detection through machine learning techniques have been a widely explore topic in the literature over the past years. However, despite the promising reported results proposed scheme are rarely deployed in production settings. This paper proposed a new technique for ML-based intrusion detection to address the high variability of network traffic and new behavior's as time passes. Our proposed model addressed network traffic behavior variability through a dynamic classifier selection scheme, while tackled new network traffic behavior through an assessment technique. As future works, we plan on extending the proposed model to incorporate into the deployed model the identified new network traffic behavior.

References

1. Andresini, G., Appice, A., Loglisci, C., Belvedere, V., Redavid, D., Malerba, D.: A network intrusion detection system for concept drifting network traffic data. In: Soares, C., Torgo, L. (eds.) Discovery Science. pp. 111–121. Springer International Publishing, Cham (2021)
2. Andresini, G., Pendlebury, F., Pierazzi, F., Loglisci, C., Appice, A., Cavallaro, L.: INSOMNIA. In: Proceedings of the 14th ACM Workshop on Artificial Intelligence and Security. ACM (Nov 2021), https://doi.org/10.1145/3474369.3486864

3. Arp, D., Quiring, E., Pendlebury, F., Warnecke, A., Pierazzi, F., Wressnegger, C., Cavallaro, L., Rieck, K.: Dos and don'ts of machine learning in computer security. In: 31st USENIX Security Symposium (USENIX Security 22). pp. 3971–3988. USENIX Association, Boston, MA (Aug 2022), https://www.usenix.org/conference/usenixsecurity22/presentation/arp
4. Chkirbene, Z., Erbad, A., Hamila, R., Mohamed, A., Guizani, M., Hamdi, M.: Tidcs: A dynamic intrusion detection and classification system based feature selection. IEEE Access 8, 95864–95877 (2020)
5. Gates, C., Taylor, C.: Challenging the anomaly detection paradigm: A provocative discussion. In: Proc. of the Workshop on New Security Paradigms (NSPW). pp. 21–29 (2006)
6. Geremias, J., Viegas, E.K., Santin, A.O., Britto, A., Horchulhack, P.: Towards multi-view android malware detection through image-based deep learning. In: 2022 International Wireless Communications and Mobile Computing (IWCMC). IEEE (May 2022), https://doi.org/10.1109/iwcmc55113.2022.9824985
7. Gu, J., Wang, L., Wang, H., Wang, S.: A novel approach to intrusion detection using SVM ensemble with feature augmentation. Computers &amp Security 86, 53–62 (Sep 2019), https://doi.org/10.1016/j.cose.2019.05.022
8. Heartfield, R., Loukas, G., Bezemskij, A., Panaousis, E.: Self-configurable cyber-physical intrusion detection for smart homes using reinforcement learning. IEEE Transactions on Information Forensics and Security 16, 1720–1735 (2021)
9. Horchulhack, P., Viegas, E.K., Santin, A.O.: Detection of service provider hardware overcommitment in container orchestration environments. In: GLOBECOM 2022 - 2022 IEEE Global Communications Conference. IEEE (Dec 2022)
10. Horchulhack, P., Viegas, E.K., Santin, A.O., Geremias, J.: Intrusion detection model updates through GAN data augmentation and transfer learning. In: GLOBECOM 2022 - 2022 IEEE Global Communications Conference. IEEE (Dec 2022), https://doi.org/10.1109/globecom48099.2022.10000666
11. Kaspersky Lab.: Kaspersky press release 2022-q3 (2022), https://www.kaspersky.com/about/press-releases/2022-hacktivists-step-back-giving-way-to-professionals-a-look-at-ddos-in-q3-2022
12. Molina-Coronado, B., Mori, U., Mendiburu, A., Miguel-Alonso, J.: Survey of network intrusion detection methods from the perspective of the knowledge discovery in databases process. IEEE Trans. on Network and Service Management 17(4), 2451–2479 (Dec 2020)
13. Moustafa, N., Turnbull, B., Choo, K.K.R.: An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things. IEEE Internet of Things Journal 6(3), 4815–4830 (Jun 2019)
14. Pinagé, F., dos Santos, E.M., Gama, J.: A drift detection method based on dynamic classifier selection. Data Mining and Knowledge Discovery 34(1), 50–74 (Oct 2019), https://doi.org/10.1007/s10618-019-00656-w
15. Salo, F., Nassif, A.B., Essex, A.: Dimensionality reduction with IG-PCA and ensemble classifier for network intrusion detection. Computer Networks 148, 164–175 (Jan 2019)
16. dos Santos, R.R., Viegas, E.K., Santin, A.O.: Improving intrusion detection confidence through a moving target defense strategy. In: 2021 IEEE Global Communications Conference (GLOBECOM). IEEE (Dec 2021)
17. dos Santos, R.R., Viegas, E.K., Santin, A.O., Cogo, V.V.: Reinforcement learning for intrusion detection: More model longness and fewer updates. IEEE Transactions on Network and Service Management pp. 1–17 (2022)
18. Sommer, R., Paxson, V.: Outside the closed world: On using machine learning for network intrusion detection. In: 2010 IEEE Symposium on Security and Privacy. IEEE (2010)
19. Tavallaee, M., Stakhanova, N., Ghorbani, A.A.: Toward credible evaluation of anomaly-based intrusion-detection methods. IEEE Trans. on Systems, Man, and Cybernetics 40(5), 516–524 (2010)
20. Viegas, E.K., Santin, A.O., Oliveira, L.S.: Toward a reliable anomaly-based intrusion detection in real-world environments. Computer Networks 127, 200–216 (Nov 2017)
21. Wahab, O.A.: Intrusion detection in the IoT under data and concept drifts: Online deep learning approach. IEEE Internet of Things Journal 9(20), 19706–19716 (Oct 2022)