Intrusion Detection Model Updates Through GAN Data Augmentation and Transfer Learning

Pedro Horchulhack*, Eduardo K. Viegas†*, Altair O. Santin*, Jhonatan Geremias*
*Pontificia Universidade Catolica do Parana (PUCPR) — Graduate Program in Computer Science (PPGIa), Brazil
{pedro.horchulhack, santin, jhonatan}@ppgia.pucpr.br
†Secure Systems Research Center — Technology Innovation Institute (TII), Abu Dhabi
eduardo@ssrc.tii.ae

GLOBECOM 2022 - 2022 IEEE Global Communications Conference | 978-1-6654-3540-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/GLOBECOM48099.2022.10000666

***Abstract*—Current machine learning techniques for network-based intrusion detection cannot handle the evolving behavior of network traffic, requiring periodic model updates to be conducted. Besides requiring huge amounts of labeled network traffic to be provided, traditional model updates demand expressive computational costs. This paper proposes a new feasible model update procedure implemented in two steps. First, we use a Generative Adversarial Network (GAN) to augment the sampled network traffic. Next, we use the augmented dataset to perform model updates through a transfer learning-based approach. Thus, our model can decrease both the number of instances that must be labeled and the computational costs during model updates. Our experiments on a one-year dataset with over 8 TB of data show that literature techniques cannot handle changes in network traffic behavior. In contrast, the proposed model without updates improved true-positive rates by up to 25.6%. With monthly model updates, it requires only 14% of computational costs and 2.3% of instances to be provided.**

***Index Terms*—Network-based Intrusion Detection, Data Augmentation, Machine Learning**

I. Introduction

Over the last years, the number of network-based cyberattacks has significantly increased and is currently on the rise. According to a cybersecurity report, the fourth quarter of 2021 has increased the number of reported network attacks by 52%, currently targeting over 15% of all Internet users [1]. In general, to detect this growing number of threats, operators resort to network-based intrusion detection systems (NIDS), through either *misuse-based* or *behavior-based* techniques [2]. The previous detects network attacks through the identification of well-known attack signatures, thus, can only detect previously known threats [3]. In contrast, the latter performs the detection task by evaluating the event behavior, signaling misconduct when a deviation from the normal modeled behavior is detected, hence, being able to signal new attacks.

Several works have been proposed for *behavior-based* intrusion detection, wherein machine learning (ML) techniques are typically used [2]. A behavioral ML model is built by evaluating a training dataset composed of a huge amount, usually millions, of labeled network traffic events from both normal and malicious players [4]. Finally, the corresponding built ML model can then be deployed in the production environment to classify network traffic samples.

Despite the high accuracies reported in the literature, *behavior-based* techniques are hardly deployed in production [5]. Networked environments pose many challenges compared to those wherein ML has been successfully applied. This is because network traffic behavior is highly variable and evolves over time, a situation that can be caused by the provision of new network services, the disclosure of new network attacks, or even changes in network communication links [6].

This kind of network traffic behavior change makes the deployed ML model outdated, as the evaluated network traffic used during the model building procedure no longer reflects the current production environment behavior [7]. As a result, the deployed ML model will significantly increase its error rates, which demand a countermeasure for executing frequent, time-consuming, and highly costly model updates [5].

Unfortunately, traditional ML model updates in NIDS are still a challenging and overlooked task, typically requiring the provision of an updated training dataset and the execution of a computationally expensive process of the entire model training process [2]. On the one hand, the operator must first recollect huge amounts of network traffic while labeling each event as either normal or attack, often through manual means, thus, demanding several weeks or even months of expert assistance. On the other hand, traditional model update techniques discard the outdated model and build a new one from scratch, increasing the needed time, and the computational costs demanded to fulfill such a task. Model updates in NIDS remain an overlooked task in the literature, wherein authors often assume that periodic model updates can be easily performed.

This paper proposes a new ML model update procedure suitable for NIDS, aimed at easing the model update task, implemented in three steps. First, the model updates are performed through a sliding window mechanism coped with a data sampling technique. The insight is to decrease the number of samples that should be labeled while also maintaining a recent collection of network traffic over time. Second, we apply a Generative Adversarial Network (GAN) for data augmentation purposes, thus, rebuilding the original network traffic distribution without demanding additional event labeling tasks to be conducted, easing the update process. Third, we perform model updates through a transfer learning approach aiming to decrease computational costs while leveraging the

knowledge of the outdated model. The main insight of our proposal is to leverage the GAN-based data augmentation to decrease the number of samples that must be labeled during model updates while also making use of transfer learning to reduce the model update computational costs.

In summary, the main contributions of this paper are:

- An evaluation of widely used ML classifiers concerning their accuracy degradation over time. Experiments performed in a dataset that spans a year with real network traffic show that current techniques significantly reduce their accuracy soon after the training period;
- A new ML model update procedure suited for network-based intrusion detection significantly decreases labeled data requirements and computational costs during model updates. The proposed model can perform the model update task with only 2.3% of training data and 14% of computational costs.

II. Preliminaries

A. Network-based Intrusion Detection

In general, a network-based intrusion detection system (NIDS) is composed of four sequential modules [8]. First, the *Data Acquisition* module collects network events for further analysis. For instance, collecting network packets from a monitored network interface card (NIC). Then, the *Feature Extraction* module performs the needed data preprocessing of the collected event, typically building a behavioral feature vector for the proper event classification.

In general, the event behavior in NIDS is represented through network flows which summarize the communication between the analyzed network entities in a given time window, e.g., the number of exchanged network packets over the last 15 seconds [9]. The built feature vector is then used as input by the *Classification* module, classifying the input as either *normal* or *attack* class. Finally, *attack*-classified events are signaled to the network operator by an *Alert* module.

Several approaches have been proposed to perform the classification task in *behavior-based* NIDS, wherein authors often resort to machine learning (ML) techniques [3]. The ML task is executed through a three-phase process, namely *training*, *validation*, and *testing*. The dataset is used to build an ML model that will be used to classify new events according to the behavior evaluated in the training dataset.

The *training* phase takes as input a training dataset composed of huge amounts, often millions, of labeled network events. During the training task, a *validation* dataset is also used to finetune the built model, such as feature selection and model parameter adjustments. Finally, the *test* phase goal is to measure the final model accuracy, which will be expected to be evidenced when the ML model is deployed in production environments.

B. Network Traffic Behavior Changes

The behavior of network traffic is a situation that the provision of new services can cause, the discovery of new attacks, or even changes in the underlying network communication link [10]. Unfortunately, these naturally occurring changes in network traffic behavior affect the deployed ML model used to detect intrusion attempts [2]. This is because the current behavior of the production environment will no longer be similar to the behavior evaluated during the *training* phase, as represented in the training dataset.

The network operator must conduct periodic model updates, a task that poses two main challenges. First, providing an updated training dataset is a challenging task in NIDS. The network operator must collect and label the network traffic, a task that is generally only achieved with expert assistance, posing high costs to be conducted. Second, a computationally expensive model training process must be performed after providing an updated training dataset.

Current approaches in the literature overlook the challenges related to model updates, neglecting how the behavior changes in network traffic may affect their proposed scheme and how model updates can be performed in an eased manner.

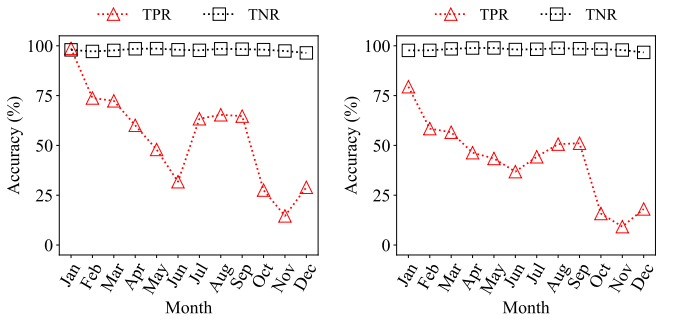
III. Related Work

Many works have been proposed for network-based intrusion detection through ML-based techniques [2]. In general, proposed approaches focus on providing higher classification accuracies. But, they neglect that their proposal will not work when there are changes in network traffic behavior. [11]. For instance, Y. Yuan *et al.* [12] makes use of an ensemble of classifiers to increase the accuracy in a widely used intrusion detection dataset, leaving model updates and changes in network traffic overlooked.

Another ensemble-based approach aiming for higher accuracies was proposed by X. Gao *et al.* [13], which considers a static dataset by applying several tree-based classifiers in cooperation. Liang and Ma [14] recently mentioned the detection rates of IDSs gradually decaying with the emergence of new attacks, which they address by downloading the latest database and retraining the whole model without previous knowledge.

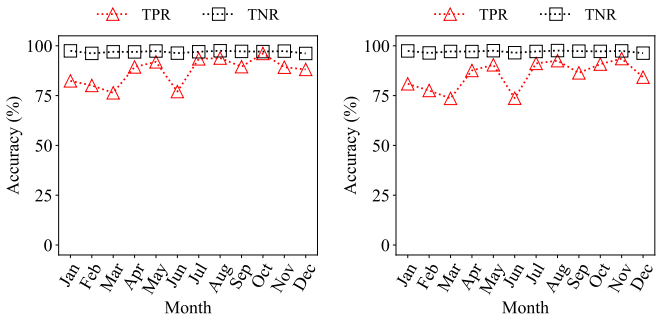
In a nutshell, network traffic behavior changes are rarely considered in the literature. However, these issues were also generated by Ying-Feng Hsu and Morito Matsuoka [15], building a controlled network traffic behavior change by applying a number of batches in a widely used intrusion detection dataset while evaluating their scheme based on the accuracy on each batch. In such a case, the changes in network traffic behavior were created in a controlled setting, therefore, it does not reproduce a realistic behavior of real-world environments.

The easiness of model updates also remains mostly overlooked in NIDS. N. Martindale *et al.* [16] proposed a stream learning intrusion detection scheme to ease the model update computational costs. However, the authors neglect how the event label can be obtained and assumes that the network traffic label can be easily requested when needed. In contrast, X. Li *et al.* [17] proposes applying the transfer learning approach to ease the model training burden in a distributed setting. The authors could ease the computational costs for model training while not addressing the difficulties related to the labeling task.



(a) Random Forest (b) Decision Tree

Fig. 1: Accuracy behavior for the selected ML classifiers in *the MAWIFlow* dataset. Classifiers were trained with January data and not updated throughout time.



(a) Random Forest (b) Decision Tree

Fig. 2: Accuracy behavior for the selected ML classifiers in *MAWIFlow* dataset. Classifiers were trained every month with 1 month worth of data.

Data augmentation techniques have been increasingly used for ML model-building purposes recently. For instance, U. Otokwala *et al.* [18] proposed a data augmentation scheme to balance the occurrence of network traffic during the model building, increasing the detection accuracy but overlooking its application for model updates. GAN-based data augmentation was proposed by G. Andresini *et al.* [19] to deal with class imbalance. The proposed model improves the detection of underrepresented classes but neglects the model update challenge.

IV. Problem Statement

In this section, we further investigate the behavior changes in network traffic and how they affect the classification performance of traditional ML-based NIDS techniques. More specifically, we introduce the used dataset and then evaluate several ML-based techniques concerning their degradation accuracy as time passes.

A. MAWIFlow Dataset

Currently, the literature assumes that network traffic behavior remains unchanged as time passes, considering that used datasets do not account for long recording periods. Consequently, proposed schemes built over such data cannot evaluate how their proposals perform when facing network traffic behavior changes.

Our work uses the *MAWIFlow* [7] dataset. The dataset was built using the Samplepoint-F from the MAWI [20] archive. As a result, it is made of real and valid network traffic that was collected daily for an interval of 15 minutes from a transit link between Japan and the USA The built dataset comprises more than 8TB of data, compounding around 4 billion in network flows. In fact, was used the whole network traffic of 2014 for evaluation purposes.r For prior event labeling, we apply an unsupervised ML technique from MAWILab [21], which automatically labels input events as *normal* or *attack*. MAWILab employs several unsupervised machine learning algorithms to find anomalies in MAWI data without individual or human assistance for the event labeling task. The anomalies are labeled as *attack*s, while the remaining data are assumed to be *normal* events. For the feature extraction task, the BigFlow [7] tool was used, which grouped events in intervals of 15 seconds while extracting 40 flow-based features from Orunada [22] work.

B. The Changes in Network Traffic Behavior

Our evaluation tests aimed at answering two main research questions: (***RQ1***) *How do ML-based techniques perform without periodic model updates?* (***RQ2***) *How do ML-based techniques perform when periodic model updates are conducted?*

Two widely used ML classifiers were selected: Decision Tree (DT) and Random Forest (RF). The DT classifier was implemented with *gini* split quality measure, and the RF classifier was implemented with 100 decision trees as its base-learner, where each one of them also uses $gini$ as the node split quality metric. A random undersampling without replacement is used in the training procedure to balance the occurrence between the classes.

The classifiers were implemented through *scikit-learn* API v0.24. The classifiers were evaluated according to their True-Positive (TP) and True-Negative (TN) rates. The TP denotes the ratio of *attack* instances correctly classified as *attack*, while the TN denotes the ratio of *normal* instances correctly classified as *normal*.

The first experiment aims at answering $RQ1$ and evaluates the classification performance of the selected classifiers when no model updates are performed over time. We train the ML model with the January data and evaluate it continuously without performing model updates.

Figure 1 shows the classification performance of the selected techniques when no periodic model updates are performed. The selected classifiers significantly reduce their classification performance after the training period. For instance, the RF classifier degrades its TP rate by 25.2% only a month after training (Fig. 1a, *Jan. vs Feb.*), while providing the worst accuracy in November, degrading its TP rate by 85.2%.

The second experiment aims at answering $RQ2$ and evaluates the classification performance of the selected techniques when periodic model updates are performed. We updated the underlying ML model at the beginning of every month with the data that occurred over the last 30 days.

Figure 2 shows the classification accuracy of the selected classifiers when periodic model updates are performed. It is possible to note that, in contrast with their no-update counterpart, the selected classifiers could provide high classification accuracies throughout time. For instance, the RF classifier provided an average TP rate of 87.3%, while its no-update counterpart reached an average TP rate of only 54.1%.

One can be noticed that periodic model updates must be performed to keep their classification accuracy as time passes. However, this is not a trivial task for NIDS, considering the challenges related to the data tagging task, as it often requires expert assistance, requiring several days or weeks of execution. To enable proper deployment of the proposed ML-based NIDS, the literature must first address the task of updating the model, allowing the operator to perform such a task without requiring significant amounts of labeled data.

V. GAN-Based Data Augmentation and Transfer Learning for Model Update Easiness

To address the above-mentioned challenge related to model updates in NIDS, we propose a GAN-based data augmentation model coped with a transfer learning scheme to ease the model update task. Our proposed approach addresses model updates threefold, as shown in Figure 3.

First, model updates are performed considering a sampled sliding window mechanism to decrease the number of samples that must be labeled. In such a case, the training data is sampled from the network traffic that was observed before the triggering of the model update task, e.g., 10% of randomly selected events over the last 7 days. The insight is that the amount of network traffic used for model updates can be significantly decreased, thus, easing the network operator labeling task if it is selected considering a sample of the sliding window of events.

Second, our proposed scheme relies on a GAN-based data augmentation technique to decrease the impact caused by the proposed network sampling approach. The rationale for such an implementation is that network traffic behavior "*gaps*" caused by the sampling approach can be recreated through a data augmentation mechanism.

Third, to decrease the computational costs during model updates, the knowledge of the outdated model is leveraged at model updates, as achieved through a transfer learning scheme. The insight of such a scheme is to significantly decrease the number of samples that must be labeled due to the sampled sliding window. It reduces the accuracy impact caused by such gaps due to the GAN-based data augmentation and also decreases the computational costs due to the transfer learning approach.

A. Classification

The proposed model assumes a traditional ML-based NIDS classification task, which relies on an ML model that uses a transfer learning scheme. The deployed ML model needs to be updated regularly to address the network traffic behavior changes over time. The network traffic is collected over a given monitored environment by a *Data Acquisition* module (Fig. 3, *Data Acquistion*). The behavior of the collected data is extracted by a *Feature Extraction* module that outputs a corresponding network flow in a feature vector format. The vector is classified as either *normal* or *attack* by a ML model, which signals misconducts to the *Alert* module.

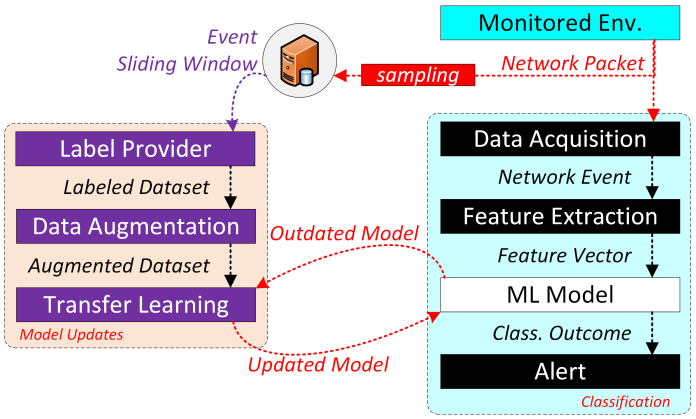


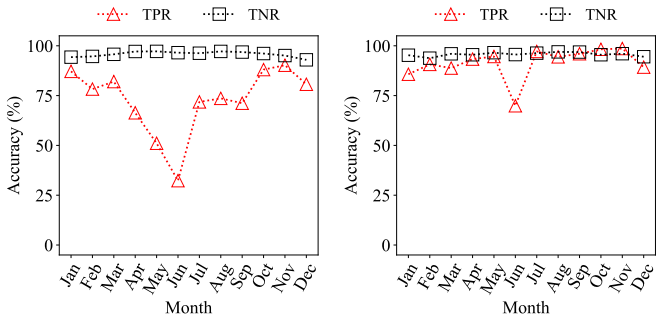
Fig. 3: Proposed GAN-based data augmentation and transfer learning mechanism for easiness model update task in NIDS.

B. Model Updates

As evaluated previously (see Section IV), the non-stationary behavior of network traffic demands periodic ML model updates be performed. However, model updates pose a significant challenge in ML-based NIDS due to the difficulties related to the data labeling task. Our proposed model relies on three main stages: a data sampling mechanism, a GAN-based data augmentation scheme, and a transfer learning approach, as shown in Figure 3. The data sampling decreases the number of samples that must be labeled by the network operator, thus, easing the model update costs. The GAN-based data augmentation aims to rebuild the sampled data's original network traffic distribution. Finally, transfer learning aims to decrease the model building computational costs during the model update, as it leverages the outdated model.

The proposed scheme insight is that we can significantly decrease the number of samples from network traffic that must be labeled by the network operator while maintaining the original network traffic distribution and reducing the computational costs of model updates. The proposed model update procedure is executed periodically, e.g., every month. In such a case, a sliding window of events, built by sampling the monitored environment behavior, must be provided (Fig. 3, *Event Sliding Window*). For instance, it randomly sampled 10% of the network events that preceded the model update task over the 7 days. The sampled dataset is then provided to the label provider module, which properly establishes each event label (Fig. 3, *Label Provider*). Several techniques can be used to fulfill such a task, including manual expert assistance or even the application of unsupervised learning techniques.

The labeled sampled dataset is used as input by the GAN-based data augmentation, which aims to rebuild the original network traffic distribution from the sampled data. The



(a) Without updates (b) Monthly model updates

Fig. 4: Behavior of the acuracy for the proposed model on the *MAWIFlow* dataset. Monthly model updates are performed with 10% of the network events sampled from network traffic.

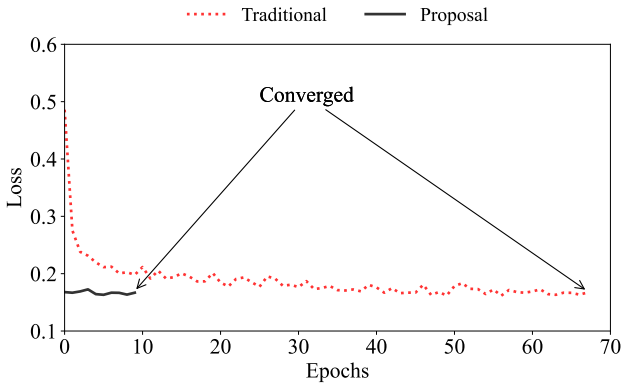
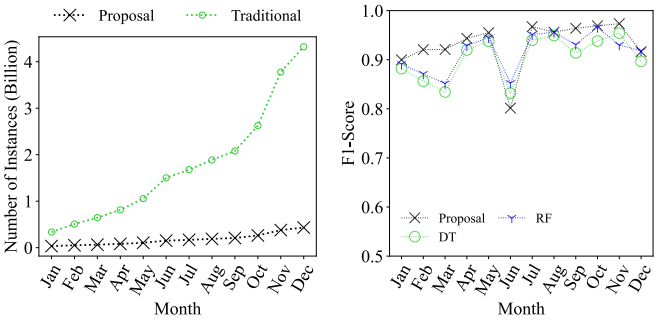


Fig. 5: Model convergence at February with and without our proposed transfer learning approach.



(a) Label request (b) F1-Score

Fig. 6: Performance of evaluated techniques with monthly model updates in *MAWIFlow* dataset over time.

augmented dataset is used for the model update procedure. The outdated model, deployed in the production environment, is retrieved and used for incremental model updates in a transfer learning approach (Fig. 3, *Transfer Learning*). Finally, the updated ML model is deployed back in the production environment.

VI. Evaluation

The evaluation of our proposed model aims at answering three main research questions: (***RQ3***) *How does our proposed model perform without periodic model updates?* (***RQ4***) *How our proposed model performs with periodic model updates?* (***RQ5***) *How does our proposed model perform when compared to traditional techniques?*

A. Model Building

Our proposed model classifier (Fig. 3, *ML Model*) was implemented and evaluated through a Multilayer Perceptron (MLP) to enable the transfer learning scheme to be applied during model updates. The MLP was implemented with 40 input features, as provided by *MAWIFlow* feature set, 512 neurons at the hidden layer, and 1 output layer. The hidden layer neurons make use of a *relu* activation function, the training relies in a learning rate of 0.001, *adam* optimizer, and $1,000$ epochs. The MLP was implemented through *scikit-learn* API v0.24.

For the GAN-based data augmentation, our proposed scheme has relied on the conditional tabular GAN (CTGAN) approach [23]. A new CTGAN is trained over the sampled data for data augmentation purposes at each model training, including the initial and periodic model updates. In such a case, 10% of augmented instances are generated and added to the sampled training dataset (Fig. 3).

B. Addressing Network Traffic Behavior Changes

The first experiment aims at answering *RQ3* and evaluates our proposed model without periodic model updates. We randomly sample 10% of network traffic events from the first week of January to achieve such a goal. The sampled data is used as input to our GAN for data augmentation purposes, while the augmented dataset is used to build the model. The model is not updated throughout the year. Therefore, we evaluate how the proposed GAN-based data augmentation can improve the model training procedure when significantly less data is provided during the model training (10% of 7 days $vs.$ 100% of 30 days).

Figure 4a shows the classification performance of our proposed model without periodic model updates. It is possible to note that our proposed technique is also affected by the network traffic behavior changes over time when no periodic model updates are performed. Despite the provision of only 10% of the original network traffic, our proposed scheme is still able to improve the average TP rates by up to 34.4% when compared to the no-update RF (Fig. 4a $vs.$ Fig. 1a). The proposed GAN-based data augmentation approach can be used to improve the lifespan of intrusion detection schemes.

The second experiment aims to answer *RQ4* and evaluate our proposed model when periodic model updates are performed. We perform monthly model updates to achieve such a goal, considering an event sliding window of 7 days and a sampling rate of 10%.

Figure 5 shows the proposed transfer learning model convergence rate compared to building the model from scratch in February. On average, the application of transfer learning during model updates decreased the computational costs by

85.3%.

Figure 4b shows the accuracy behavior of our proposed model with monthly model updates. It is possible to note that our proposed technique provided significantly high accuracy rates, presenting an average of 91.4%, and 95.7% of TP and TN rates, respectively, an improvement of 25.6% in the TP rate when compared to its no-update counterpart.

Our proposed technique significantly eased the model update task, demanding only 14% of computational costs while using only 2.3% of the network events. This is because it uses only 10% of randomly sampled data from the last 7 days, while the traditional techniques were trained with 100% of the last 30 days (Fig. 4b vs. Fig. 2).

To answer *RQ5* we further investigate our proposed model benefits compared to traditional techniques. Figure 6a shows the cumulative number of network samples that the evaluated techniques must label during model updates.

Our proposed model demanded only 2.3% of samples to be labeled compared to traditional approaches, posing a significantly easier to execute model update task. Figure 6b shows the classification accuracy of our proposed model versus the techniques evaluated previously (Fig. 4b vs Fig. 2). The approaches were evaluated according to their F1-Score, computed as the harmonic mean of precision and recall values. Our proposed model was able to provide similar F1-Scores when compared to traditional approaches while demanding only 2.33% of labeled network events to be delivered and 14% of computational costs during model updates.

VII. Conclusion

Network traffic behavior changes are a known and overlooked challenge to ML-based NIDS. This paper has shown that current approaches in the literature cannot keep their classification accuracy for long periods, demanding frequent and challenging model updates to be conducted.

Our proposed model was able to significantly ease the model update task through a sampled sliding window mechanism coped with GAN-based data augmentation and the application of transfer learning over the outdated deployed ML model.

Acknowledgment

This work was partially sponsored by Brazilian National Council for Scientific and Technological Development (CNPq) grant nº 304990/2021-3.

References

[1] *Kaspersky Reports DDoS attacks in Q4 2021*, February 2022. [Online]. Available: https://securelist.com/ddos-attacks-in-q4-2021/105784/

[2] B. Molina-Coronado, U. Mori, A. Mendiburu, and J. Miguel-Alonso, "Survey of network intrusion detection methods from the perspective of the knowledge discovery in databases process," *IEEE Trans. on Network and Service Management*, vol. 17, no. 4, pp. 2451–2479, Dec. 2020.

[3] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *2010 IEEE Symposium on Security and Privacy*, 2010, pp. 305–316.

[4] R. R. dos Santos, E. K. Viegas, and A. O. Santin, "A reminiscent intrusion detection model based on deep autoencoders and transfer learning," in *2021 IEEE Global Communications Conference (GLOBECOM)*. IEEE, Dec. 2021.

[5] P. Horchulhack, E. K. Viegas, and A. O. Santin, "Toward feasible machine learning model updates in network-based intrusion detection," *Computer Networks*, vol. 202, p. 108618, Jan. 2022.

[6] C. Gates and C. Taylor, "Challenging the anomaly detection paradigm: A provocative discussion," in *Proceedings of the 2006 Workshop on New Security Paradigms*, ser. NSPW '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 21–29.

[7] E. Viegas, A. Santin, A. Bessani, and N. Neves, "BigFlow: Real-time and reliable anomaly-based intrusion detection for high-speed networks," *Future Generation Computer Systems*, vol. 93, pp. 473–485, Apr. 2019.

[8] B. B. Bulle, A. O. Santin, E. K. Viegas, and R. R. dos Santos, "A host-based intrusion detection model based on OS diversity for SCADA," in *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, Oct. 2020.

[9] N. Williams, S. Zander, and G. Armitage, "A preliminary performance comparison of five machine learning algorithms for practical ip traffic flow classification," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 5, p. 5–16, Oct 2006.

[10] V. Abreu, A. O. Santin, E. K. Viegas, and V. V. Cogo, "Identity and access management for IoT in smart grid," in *Advanced Inf. Net. and App.* Springer International Publishing, 2020, pp. 1215–1226.

[11] E. K. Viegas, A. O. Santin, V. V. Cogo, and V. Abreu, "A reliable semi-supervised intrusion detection model: One year of network traffic anomalies," in *IEEE Int. Conf. on Comm. (ICC)*, 2020, pp. 1–6.

[12] Y. Yuan, L. Huo, and D. Hogrefe, "Two layers multi-class detection method for network intrusion detection system," in *2017 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, Jul. 2017.

[13] X. Gao, C. Shan, C. Hu, Z. Niu, and Z. Liu, "An adaptive ensemble machine learning model for intrusion detection," *IEEE Access*, vol. 7, pp. 82 512–82 521, 2019.

[14] J. Liang and M. Ma, "Co-maintained database based on blockchain for idss: A lifetime learning framework," *IEEE Transactions on Network and Service Management*, pp. 1–1, 2021.

[15] H. Benaddi, K. Ibrahimi, A. Benslimane, and J. Qadir, "A deep reinforcement learning based intrusion detection system (DRL-IDS) for securing wireless sensor networks and internet of things," in *Lecture Notes of the Institute for Computer Sciences*, 2020, pp. 73–87.

[16] N. Martindale, M. Ismail, and D. A. Talbert, "Ensemble-based online machine learning algorithms for network intrusion detection systems using streaming data," *Information*, vol. 11, no. 6, p. 315, Jun. 2020.

[17] X. Li, Z. Hu, M. Xu, Y. Wang, and J. Ma, "Transfer learning based intrusion detection scheme for internet of vehicles," *Information Sciences*, vol. 547, pp. 119–135, Feb. 2021.

[18] U. Otokwala, A. Petrovski, and H. Kalutarage, "Improving intrusion detection through training data augmentation," in *International Conference on Security of Information and Networks (SIN)*. IEEE, Dec. 2021.

[19] G. Andresini, A. Appice, L. D. Rose, and D. Malerba, "GAN augmentation to deal with imbalance in imaging-based intrusion detection," *Future Generation Computer Systems*, vol. 123, pp. 108–127, Oct. 2021.

[20] MAWI, "MAWI Working Group Traffic Archive - Samplepoint F," 2021. [Online]. Available: https://mawi.wide.ad.jp/mawi/

[21] R. Fontugne, P. Borgnat, P. Abry, and K. Fukuda, "MAWILab: Combining diverse anomaly detectors for automated anomaly labeling and performance benchmarking," in *Proc. of the 6th Int. Conf. on emerging Networking EXperiments and Technologies (CoNEXT)*, 2010.

[22] J. Dromard, G. Roudiere, and P. Owezarski, "Online and scalable unsupervised network anomaly detection method," *IEEE Trans. on Net. and Service Management*, vol. 14, no. 1, p. 34–47, Mar 2017.

[23] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling tabular data using conditional gan," in *Advances in Neural Information Processing Systems*, 2019.