A Multi-View Intrusion Detection Model for Reliable and Autonomous Model Updates

Rivaldo L. Tomio*, Eduardo K. Viegas*, Altair O. Santin*, Roger R. dos Santos*
*Graduate Program in Computer Science
Pontifical Catholic University of Parana, Brazil
{rivaldo.luiz, eduardo.viegas, santin, robson.roger}@ppgia.pucpr.br

***Abstract*—Changes in network traffic behavior over time are neglected by authors who use machine learning techniques applied to intrusion detection. In general, it is assumed that periodic model updates are performed, regardless of the challenges related to such a task. This paper proposes a new multi-view intrusion detection model capable of reliably performing model updates without human assistance while also maintaining its accuracy over time. The proposal evaluates the classification's confidence values in a multi-view configuration to maintain its reliability over time, even without model updates. Besides, it is able to perform model updates autonomously, according to the result of the multi-view classification. Our experiments, performed with 7TB of real network traffic over a 2-year interval, show that our proposed scheme can maintain its accuracy over time without model updates, rejecting only 14.2% of its classification. However, when autonomous model updates are performed, the rejection rate drops to just 8.8%, while also improving the model's accuracy by 4.3%.**

***Index Terms*—Intrusion Detection, Machine Learning, Multi-View, Model Updates.**

I. Introduction

In the second quarter of 2020, Kaspersky solutions have identified over 800 million network attacks [1]. As a result, operators need access to security solutions to detect this growing number of threats [2]. In general, administrators often employ Network-based Intrusion Detection Systems (NIDS) to detect malicious activities in network traffic through either *misuse-based* or *behavior-based* approaches [3]. The previous searches for attack footprints, namely signatures, within the network traffic, thus, can only detect well-known attacks. On the other hand, *behavior-based* techniques aim to detect attacks by analyzing the network traffic behavior. As a result, it can detect new attacks, as long as their behavior is similar to known threats or significantly differs from benign ones [2].

In the literature, *behavior-based* NIDS has often been performed through machine learning (ML) techniques, wherein pattern recognition approaches are typically used. This kind of detection scheme relies on a training dataset to extract a behavioral model. Then, the built model can be used for the classification of further events [4]. Consequently, if the environment behavior changes, e.g., a new attack is discovered or a new service is provided, the underlying ML model becomes unreliable. This is because the training dataset in which the ML model was built does not contain the current environment behavior, rendering the ML model outdated [5].

Obsolete ML models are unable to reach the same level of accuracy as those measured during the test phase. Thus, due to an increase in the error rate over time, operators will often discard further alarms [6].However, despite the changes in network traffic behavior over time being a known issue in NIDS field, such a challenge is often neglected in related works [2]. In the literature, the majority of proposed ML-based intrusion detection schemes pursue higher classification accuracies, paying little or no attention to the challenges involved during the model update task [7]. In contrast, researchers often assumes that the network traffic is static, and no changes occur over time, or even that periodic model updates are performed, without even evaluating the need for such updates. Ideally, the ML model must be as recent as possible, taking into account that network traffic behavior may change drastically in a small period, rendering it unreliable [6]. However, the model retraining task is a computationally expensive process that often demands human assistance for the labeling of events, e.g., tag network traffic as either normal or attack, which is not always available or is available with a high cost [8]. As a result, the ML model update task remains overlooked in the literature, and even the lifespan of widely used ML-based techniques is yet to be known [9].

This paper proposes a novel multi-view intrusion detection model aiming for autonomous model updates and higher detection reliability over time, in a twofold way. First, we leverage a multi-view technique to address the lack of reliability in network traffic classification over time. We assume that we can improve the system reliability when using several distinct and complementary feature sets, namely views. Our insight is that the classification confidence can be used to measure reliability in classification, wherein each view in a multi-view procedure can be used to improve classification reliability even with outdated models. Second, we provide an autonomous model update scheme that leverages the proposed reliable multi-view approach to label network traffic for the model update procedure, thus operating without human assistance. In summary, the main contributions of our paper are:

- We evaluate commonly used ML-based intrusion detection approaches concerning their classification reliability over time. Experiments performed with a 2-year long labeled network traffic, composed of over 7TB of data, have shown that current approaches in the literature

significantly decreases their accuracy months after the training period, regardless of their used feature sets;
- We propose and evaluate a novel multi-view intrusion detection model that can autonomously withstand reliably for long periods of time even without model updates. If so, our proposal can further increase detection accuracy while rejecting fewer events without human assistance;

II. Preliminaries

A. Network-based Intrusion Detection

Network-based Intrusion Detection Systems (NIDS) aims at finding malicious activities within a network environment [7]. A typical NIDS architecture is comprised of four sequential modules, namely *data acquisition, feature extraction, decision* and *alert*. The *data acquisition* module collects data from the environment, reading network packets from a NIC (network interface card). The *feature extraction* module extracts behavioral features from the network data to compound a feature vector, also named as view, e.g. summarizes network packets in a 15-s interval network flow. Finally, the *decision* module establishes the proper event label from the feature vector evaluation e.g., applies an ML model to classify it as normal or attack for the *alert* module proper report it.

In general, in production environments, *signature-based* detection techniques are used by NIDS tools [2]. Such an approach searches the collected data for known attack patterns. Recently significant research effort has been conducted in developing novel attack detection techniques in NIDS, typically through *behavior-based* approaches.

B. Machine Learning for NIDS

The majority of *behavior-based* detection approaches in NIDS relies on machine learning (ML) techniques, typically through pattern recognition means [6]. In such a case, a behavioral model is built through a training dataset, which comprises the expected network traffic behavior for a given period. The training procedure is often performed in a supervised setting, wherein the network event label must be previously known, as the ML model will be built taking it into consideration. However, networked environments present a plethora of challenges not commonly evidenced in other fields [2].

Network traffic behavior constantly changes, either due to new types of services or due to new attacks being discovered [7]. These changes in network traffic behavior over time render the built ML model outdated, which may increase the error rate. However, ML-based techniques are often designed for finding similarities in its input data, rather than finding not previously seen behavior [6]. As a result, if the occurrence of new network traffic, a significant change in the network traffic behavior, as represented by its feature set (view), the built ML model may significantly decrease its accuracies, rendering it unreliable.

Conduct a model update procedure in networked environments is not an easily achieved task, given that the network traffic must be labeled, typically with human assistance, and a computationally expensive process of model retraining must be performed. As a result, despite the reports of several works regarding highly accurate ML models, such techniques remain mostly a research topic, hardly being deployed in production environments [2] [6].

III. Related Works

In the literature, authors often aim at providing higher detection accuracies in a single and static dataset. For instance, Mirza *et al.* [13] proposes an ensemble of classifiers to improve their system detection accuracy. The authors assign a classifier weight according to their measured accuracies, wherein more accurate models have higher classification influence. Singh *et al.* [14] proposes a Random Forest classifier with a K-Means in a hybrid approach for intrusion detection. The authors demonstrated that, in this way, they could maintain high detection rates while also decreasing false alarms. However, although high accuracy rates are reported, these works rely on outdated datasets, through KDD99 [15], that do not consider the natural variability of network traffic behavior over time.

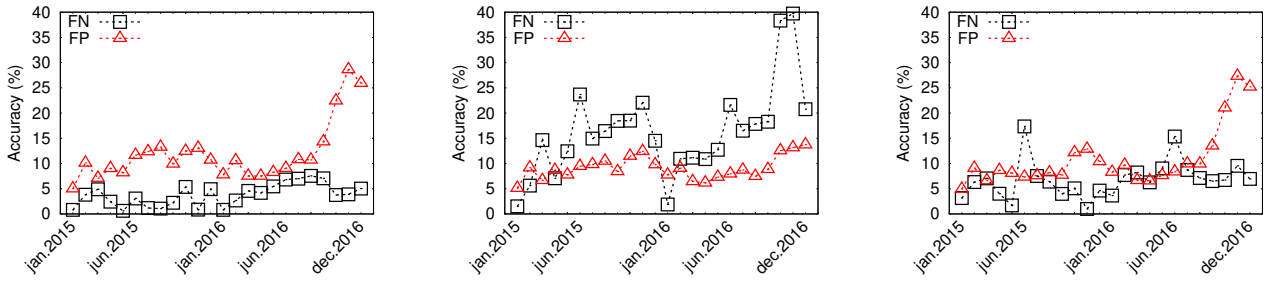
As another approach to improve accuracy, some authors resort to multi-view detection techniques by leveraging distinct and complementary feature sets. For instance, He *et al.* [16] proposes a multimodal approach as a multi-view technique for intrusion detection. The authors are able to significantly improve detection accuracy on a variety of datasets, including outdated and novel ones. However, the authors do not take into account the network traffic behavior changes over time. On the other hand, Li *et al.* [17] proposes a multi-view approach for spam detection in resource-constrained environments. The authors assume a semi-supervised setting wherein a multi-view setting is used for the label of other events. Although the authors consider a more realistic scenario with model updates, their used dataset does not present a long data period without natural behavior changes.

To address the model update challenge, authors often assume that periodic model updates can be performed. For instance, Xiao *et al.* [18] applies an online random forest to detect disk failures over time. However, the authors assume that the event label can be provided when needed, an unrealistic NIDS assumption. Doak *et al.* [19] proposes a self-updating model with error remediation. Their proposed scheme is able to update their model over time autonomously. However, their technique demands the periodic evaluation of their underlying error rates, the proper event label, also not feasible in NIDS.

To the best of our knowledge, we are the first to address the model update challenge in a realistic setting in NIDS. That is, autonomously and reliably updating the underlying ML models through a multi-view classification scheme.

IV. Problem Statement

The changes in network traffic behavior over time is a known and overlooked challenge in NIDS literature. The lifespan of proposed intrusion detection schemes remains unknown. The accuracy degradation caused by the natural



(a) Random Forest - Nigel [10] features. (b) Random Forest - Orunada [11] features. (c) Random Forest - Viegas [12] features.

Fig. 1: Accuracy behavior in a 2-year interval with a Random Forest classifier varying its used feature sets (view). Classifier is trained in January 2015 and evaluated in the remaining period without updates.

changes in network traffic as time passes is not even evaluated. This section further investigates how traditional ML-based techniques perform while detecting network-related intrusion attempts for long periods. We first introduce our novel intrusion dataset that contains real and labeled network traffic for a 2-year interval. Then, we evaluate a widely used ML technique concerning the accuracy degradation over time, with several feature sets from related works.

A. MAWIFlow Intrusion Dataset

Widely used datasets are even decades old, containing unrealistic network traffic, outdated attacks, and several known flaws [15]. In principle, datasets used for such purposes must contain real and valid network traffic, containing the proper communication from services and attacks experienced in the wild. However, the monitoring and labeling of real network traffic for NIDS design purposes, e.g., record the network traffic from a university, is not possible due to possible privacy concerns.

Our work proposes the *MAWIFlow* dataset, a publicly available intrusion dataset containing real, valid, and labeled network traffic from production environments that span for an extended period. To provide such characteristics, *MAWIFlow* is built on top of MAWI [20] working group traffic archive. It contains the network traffic from MAWI samplepoint-F, a transit link between Japan and the USA collected for a 15-seconds long interval daily. For this work, the network data from a 2-year interval was used, from 2015 to 2016.

The network data is collected daily, containing a network PCAP file for each day throughout the evaluated 2-year interval, comprising over 7TB of data and over 70 billion network flows. The collected network data is summarized in network flows according to the hosts and services involved in each communication. Each network flow comprises 15-sec of client/service and server/service data, which is then summarized in a set of feature vectors (views).

The built dataset, namely *MAWIFlow*, is made of 3 distinct views, extracted for each network flow, namely Nigel [10], Orunada [11], and Viegas [12], each composed by 20, 15 and 47 features respectively. Hence, each feature set provides a distinct view for the same event, represented by the extracted set of features. For labeling purposes, our work applies MAWILab [21] unsupervised ML algorithms that identify network anomalies, which are then labeled as attacks in our data.

B. View Performance Over Time

We evaluate the accuracy degradation of the ML algorithm with respect to the underlying used view. It is important to note that regardless of the ML classifier, the feature set used for the classification is the defining aspect of the ML classification reliability. This is because if a network traffic behavior change occurs, it will only affect the ML classification accuracies, hence, its reliability, if the extracted features are also affected, as the ML model classifies events according to its input feature set values. Therefore, our evaluation further investigates the impact of the natural changes in network traffic to the used ML algorithm, according to distinct views commonly used in related works [11] [12] [10].

We apply the Random Forest algorithm, a widely used ML classifier, with 100 decision trees as its base learner, using the first month of January as the training dataset for the random forest and evaluated each view separately throughout the year without model updates. Due to the imbalanced nature of the dataset, most events are normal (around 99%). We conducted a random undersampling without a replacement stratification procedure in the training dataset to balance the class occurrences while implemented classifiers using the $scikit-learn$ API. The classifiers were evaluated according to their False-Positive (FP) and False-Negative (FN) rates. The FP denotes the ratio of normal events misclassified as attacks, while the FN denotes the ratio of attack events misclassified as benign ones.

Figure 1 shows the monthly measured error rates according to each built random forest classifier, with Nigel [10], Orunada [11], and Viegas [12] feature sets. It is possible to note an error rate increase in the months that followed the training period (January 2015), regardless of the underlying used feature set. However, the error rate increase varies according to the underlying ML view. For instance, Nigel [10] view increases its FN rate throughout 2015 while maintaining its FP rates stable meanwhile, while Orunada [11] view significantly

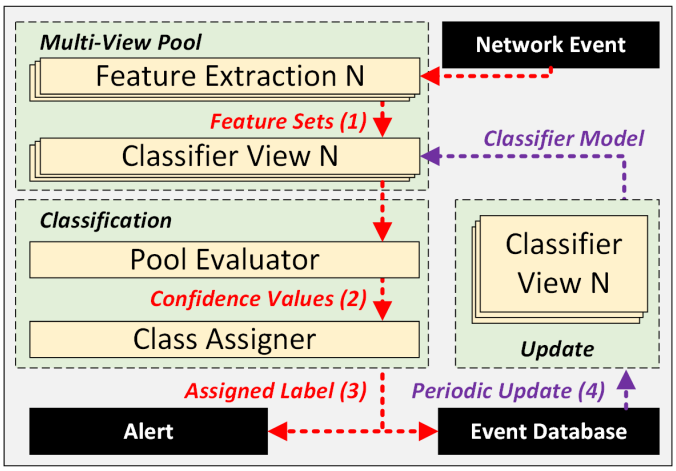


Fig. 2: Proposed multi-view intrusion detection model for reliable and autonomous model updates.

increases both FP and FN rates in 2015 while provides more accurate results in 2016.

On average, the built classifiers increased their FP and FN rates by 3.7% and 4.8% in the first month after the training period. The worst accuracy decrease was noted by all used views in November 2016, as either noted by an increase in their FP or FN rates. Therefore, regardless of the used feature set, the natural changes over time in network traffic affects their reliability. However, each used feature set presented a different impact on their reliability over time, thus showing that the used view can help improve the system reliability if the proposed mechanism can explore such property.

V. A Multi-View Intrusion Detection Model

We present a novel multi-view intrusion detection model to address the evolving behavior of network traffic evaluated previously, composed by two main steps (*Classification* and *Autonomous Model Update*, as shown in Figure 2) and aims to ensure, thus maintain, the system accuracy over time, and also perform autonomous model updates, without human assistance.

The proposal considers a multi-view classification scheme, wherein a pool of ML models are used, each model is built through a distinct set of features, namely view (such as those evaluated in Figure 1). The classification procedure starts with a networking event for classification, e.g., a set of network packets related to service occurred within a time interval. The network events' behavior is then extracted by a set of feature extraction modules, in which each module extracts a distinct feature set. The computed feature sets are forwarded for classification, where each classifier, with its own view, outputs a classification confidence value. The classification confidence values are used as a measure of classification correctness by the pool evaluator, which its goal is to only accept highly confident classifications as an attempt to maintain its reliability over time, even with outdated models. Therefore, only highly confident classifications, as output by distinct classifiers, each with a unique event view, is used for the event labeling process, thus, discarding possible outdated classifications and most likely errors.

As an attempt to provide up-to-date ML models, our proposal performs periodic autonomous model updates according to the assigned event label, as obtained by the most confident views. Therefore, a view that is currently producing low confident classifications can be reliably updated by other complementary views, which are outputting higher confidence values, and more likely accurate classifications, thus, autonomously and reliably updating the ML models over time. The next subsections further describes the proposed multi-view intrusion detection model, including the *classification* and *autonomous model update* modules.

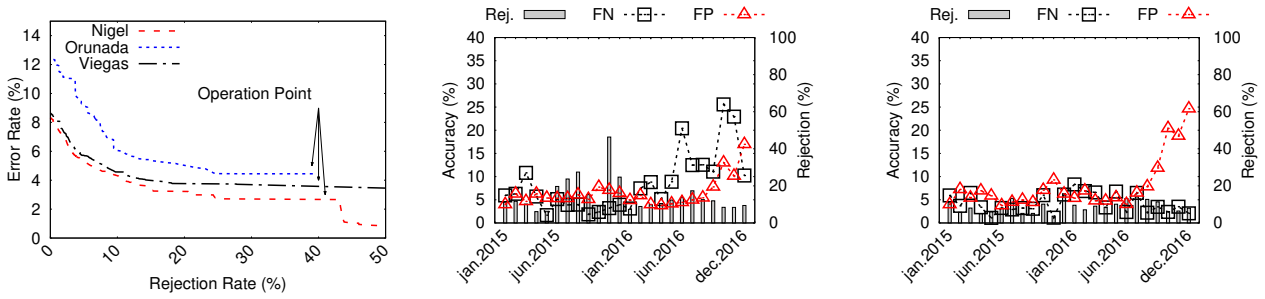
A. Reliable Multi-View Classification

A multi-view approach relies on two primary principles (i) consensus assumes that all views should maximize the agreement on multiple distinct views, while (ii)complementary assumes that each view may contain some knowledge that other views do not have. Our proposal explores such characteristics as an attempt to improve system reliability, even in the face of new network traffic. Our main assumption is that complementary views can improve one another over time, during both classification and autonomous model update.

The reliable multi-view classification approach is performed in a twofold manner. First, the behavior of the network event to be classified is extracted by a set of feature extraction pool, which outputs a set of feature sets, namely view (*Feature Sets*, Figure 2). The extracted feature sets are classified by a set of classifiers, where each model outputs a classification confidence value (*Confidence Values*, Figure 2). The classification confidence value is classifier agnostic, for instance, the Random Forest classifier outputs its classification confidence values according to the ratio of its base-learners that classified the instance as the assigned label. The confidence values are used by a pool evaluator module to identify confident classifications that should be used for the label assignment process. Identifying confident classifications should be defined according to the operator's needs, for instance, to improve reliability, one can use a higher confidence threshold despite an increase in the rejection rate. The rejection is measured as a ratio of events that did not meet the established confidence threshold and should have their alerts suppressed to maintain the system reliability over time. Finally, the class assigner establishes the event label through a majority voting procedure from the accepted classifications, i.e., classification views that met the used classification confidence threshold.

B. Autonomous Model Update

Regardless of the evaluation of the classification correctness performed during the classification task, the underlying ML models will become further unreliable as time passes. Therefore, an increase in the rejection rate will be experienced, caused by the lack of an updated ML model, as they will be unable to cope with the current environment behavior.



(a) Error-reject tradeoff for each evaluated feature set (view) in February 2015.

(b) Accuracy and rejection rates over time without model updates..

(c) Accuracy and rejection rates over time with autonomous model updates.

Fig. 3: Proposed reliable multi-view intrusion detection obtained accuracy and rejection rates over time in MAWIFlow dataset.

However, perform periodic model updates is not an easily achieved task, as it demands expert assistance to provide the proper event label. Our proposal performs autonomous model updates without human assistance by leveraging the multi-view pool to address such a challenge.

The update procedure is performed periodically, for instance, every semester. At each update task, the event database (Figure 2) is used for the pool update procedure. The event database is composed of the network events, autonomously labeled by the reliable multi-view classification pool (Section V-A). Thus, the event labels are assigned according to the most confident views and, consequently, are used to improve other less confident views at each model update. As a result, our proposal can improve the model views at each model update procedure, provide up-to-date ML models over time, and reliably and autonomously update the underlying ML models.

VI. Evaluation

The evaluation aims at answering the following research questions: ($Q1$) *Can the evaluation of the classification confidence value be used to improve the ML model accuracy?* ($Q2$) *Is the proposed pool evaluator able to maintain the system reliability over time, without model updates?* ($Q3$) *Is the proposed scheme able to update the ML model and remain reliable over time autonomously?*

A. Model Building

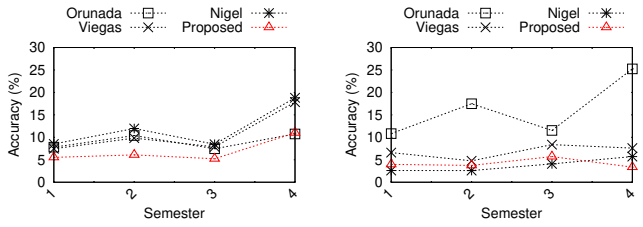
Our proposed reliable multi-view intrusion detection model was built, taking into account the three previously evaluated views (Figure 1). Therefore, three random forest classifiers were trained, each with a unique event view, using Nigel [10], Orunada [11], and Viegas [12] views, respectively. Similarly, the random forest classifiers were trained with 100 decision trees as their base-learners. The confidence values, used by our proposal to ensure reliability (Section V-A), was obtained through the scikit-learn API, as computed by the $predict\_proba$ API function. The API computes the random forest classifier's confidence values as the ratio of individual trees that classified the evaluated instance as the outputted class.

B. Multi-View for Long-term Classification Reliability

The first experiment aims at answering question $Q1$ and compares each view error-reject tradeoff. Each classifier is trained in January 2015 and evaluated in February 2015. The classification thresholds were defined through the Class-Related-Threshold (CRT) [7] approach. The computed confidence values for each individual view in February are evaluated to establish the error-reject tradeoff. It enables to define the set of thresholds that should be used throughout the remaining 2-year interval (confidence thresholds, Section V-A). For all evaluated views, shown in Figure 3a, one can see that it is possible to decrease the error rate with an increase in the rejection rate as a tradeoff. Therefore the confidence can be used to assess the classification quality. Besides, each view provides different tradeoffs, with the Viegas [12] feature set, providing the best relation between error and rejection.

The second experiment aims to answer question $Q2$ and apply the obtained classification thresholds in $Q1$ throughout the remainder 2-year interval. To achieve such a goal, each view classification threshold is set at a 40% rejection rate as measured in February 2015 (*Operation Point*, Figure 3a). The defined set of thresholds for each view is used throughout the remaining 2-year interval without model updates. Thus, if a classification in one view does not meet the defined class-related threshold, it is not used for the pool majority voting process, and if the classification outcome from all views also did not meet their thresholds, the event is rejected.

Figure 3b shows the proposal obtained accuracy and rejection rates over time without updates when applying the defined classification thresholds (Figure 3a) throughout *MAWIFlow* dataset 2-year interval. One can be noticed that despite using a 40% rejection rate operation point, the proposal is able to reject significantly fewer instances, rejecting an average of 14.2% instances in the dataset. Over time, the accuracy rates also remain further stable compared to their training period (January 2015), increasing their FP on average by only 2.1% while maintaining its FN rate stable in 2015. Thus, when compared to the traditional one-view approach (Figure 1), the proposed multi-view method is able to remain reliable for more extended periods, increasing their FP on average by only 3.5% in the 2-year interval, in contrast to the one-



(a) FP rates over time. (b) FN rates over time.

Fig. 4: Proposed and traditional approaches error rates.

view technique, that increases its FP by 9.2%, 6.2%, and 8.0% with the Nigel [10], Orunada [11] and Viegas [12] views respectively.

Finally, to answer question $Q3$, we autonomously update the underlying ML models over time, at a semester basis (6-month interval), according to the assigned event label, without human assistance. Figure 3c shows the proposal obtained accuracy and rejection rates with model updates in a 6-month interval basis when applying the defined classification thresholds (Figure 3a) throughout *MAWIFlow* dataset 2-year interval. The proposed scheme significantly decreases the rejection rate while also improving its accuracy, when autonomous model updates are performed, rejecting in average only 8.8%, in contrast to 14.2% if no model updates are performed while improving its accuracy by 4.3%.

Figure 4 shows a comparison of our proposed multi-view approach to a single-view one with respect to their accuracy in MAWIFlow dataset. The proposed model provided the lowest error rates over time, reaching at least, or even improving, the accuracy rates to the best view in each semester, without significantly decreasing its accuracy, as occurs in the single view approaches. On average, the proposed model maintained the same FN rate while only increasing its FP rates by only 4.8% throughout the 2-year interval. In contrast, the single view technique increased their FP by 9.2%, 6.2%, and 8.0% and also their FN by 0.1%, 11.9%, and 1.9% with the Nigel [10], Orunada [11] and Viegas [12] views respectively in the same interval.

VII. Conclusion

Despite the plethora of works that report highly accurate ML-based intrusion detection schemes, proposed techniques are hardly used in production. ML-based intrusion detection faces a broader range of challenges than those where it has been successfully applied, one of which is related to the network traffic behavior changes over time. This paper has proposed and evaluated a novel multi-view approach for autonomous and reliable model updates. Our proposed scheme reliably updated the underlying ML models by leveraging a multi-view and an evaluator technique. The evaluator assesses the classification quality to ensure that only highly confident classifications are accepted. On the other hand, the multi-view ensures that the system remains reliable over time by using each view to improve other complimentary views during both classification and model updates.

Acknowledgment

This work was partially sponsored by Brazilian National Council for Scientific and Technological Development (CNPq), grant nº 430972/2018-0).

References

[1] *IT threat evolution Q2 2020*, September 3, 2020. [Online]. Available: https://securelist.com/it-threat-evolution-q2-2020-pc-statistics/98292/

[2] 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*. IEEE, 2010.

[3] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.

[4] M. Ring *et al.*, "A survey of network-based intrusion detection data sets," *Computers & Security*, vol. 86, pp. 147–167, 2019.

[5] 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.

[6] C. Gates and C. Taylor, "Challenging the anomaly detection paradigm: A provocative discussion," in *Proc. of the Workshop on New Security Paradigms (NSPW)*, 2006, pp. 21–29.

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

[8] R. R. dos Santos, E. K. Viegas, A. Santin, and V. V. Cogo, "A long-lasting reinforcement learning intrusion detection model," in *Advanced Information Networking and Applications*, 2020, pp. 1437–1448.

[9] E. K. Viegas, A. O. Santin, and V. Abreu, "Machine learning intrusion detection in big data era: A multi-objective approach for longer model lifespans," *IEEE Trans. on Network Science and Engineering*, 2020.

[10] 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.

[11] 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.

[12] E. Viegas, A. O. Santin, A. Franca, R. Jasinski, V. A. Pedroni, and L. S. Oliveira, "Towards an energy-efficient anomaly-based intrusion detection engine for embedded systems," *IEEE Transactions on Computers*, vol. 66, no. 1, p. 163–177, Jan 2017.

[13] A. H. Mirza, "Computer network intrusion detection using various classifiers and ensemble learning," in *2018 26th Signal Processing and Communications Applications Conference (SIU)*. IEEE, May 2018.

[14] P. Singh and M. Venkatesan, "Hybrid approach for intrusion detection system," in *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*. IEEE, Mar. 2018.

[15] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, Jul 2019.

[16] H. He, X. Sun, H. He, G. Zhao, L. He, and J. Ren, "A novel multimodal-sequential approach based on multi-view features for network intrusion detection," *IEEE Access*, vol. 7, pp. 183 207–183 221, 2019.

[17] W. Li, W. Meng, Z. Tan, and Y. Xiang, "Design of multi-view based email classification for IoT systems via semi-supervised learning," *Journal of Net. and Comp. App.*, vol. 128, pp. 56–63, Feb. 2019.

[18] J. Xiao, Z. Xiong, S. Wu, Y. Yi, H. Jin, and K. Hu, "Disk failure prediction in data centers via online learning," in *Proc. of the 47th International Conference on Parallel Processing*. ACM, Aug. 2018.

[19] J. E. Doak, M. R. Smith, and J. B. Ingram, "Self-updating models with error remediation," in *Art. Int. and Machine Learning for Multi-Domain Op. App. II*. SPIE, May 2020.

[20] MAWI, "MAWI Working Group Traffic Archive - Samplepoint F," 2020. [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.