# Temporal Representations for Detecting BGP Blackjack Attacks

Rafael Copstein
*Faculty of Computer Science*
*Dalhousie University*
Halifax NS, Canada
rafael.copstein@dal.ca

Nur Zincir-Heywood
*Faculty of Computer Science*
*Dalhousie University*
Halifax NS, Canada
zincir@cs.dal.ca

***Abstract*—Even though BGP blackholes are used to mitigate denial of service attacks, they also represent a major cybersecurity challenge to organizations. These challenges include abuse of route selection algorithms, lack of host verification, and maliciously triggering a blackhole, i.e. BGP blackjack. This research presents a supervised machine learning based approach for blackjack detection. We employ Naive Bayes and Decision Tree classifiers with three different temporal representations: (i) packets with/without timestamps; (ii) buffer of packets with/without timestamps; and (iii) overlapping / non-overlapping buffer of packets with/without timestamps. Our goal is to understand the effect of temporal data and context in the detection of blackjack attacks. Furthermore, we explore the most suitable attributes and solution complexity. Evaluations show that using overlapping buffer data with timestamps achieves the highest accuracy/recall using five of the seven BGP attributes. We also observe that high performance is not correlated with complex solutions.**

***Index Terms*—BGP, blackholing, blackjack attacks, security**

## I. Introduction

Distributed denial of service (DDoS) is a popular form of attack with the objective of taking down services on the Internet. One of the techniques that emerged to mitigate this kind of attack is called BGP Blackholing, i.e. dropping traffic to a destination (prefix). The Border Gateway Protocol (BGP) [1] is the Internet's *de facto* wide area network (interdomain) routing protocol. Using BGP update messages, a BGP blackhole can be requested by an Autonomous System (AS) for a certain IP prefix by announcing a route to peer ASes using BGP communities [2]. If the blackhole is established, all traffic destined to the requested IP prefix is dropped, sparing the host machine from the attack at the cost of suffering some (but possibly not complete) unavailability.

As seen in [3], most blackholes last at least as much as the attacks themselves, which have been observed to have a duration of 10 minutes or less in almost half the cases. However, this can be longer given that almost half of the observed cases took one hour or more to have their blackholes deactivated after the end of an attack [3]. Being built on top of BGP, blackholing does not have a host verification mechanism in place apart from the peer AS's number. This results in security vulnerabilities [4], making ASes that support BGP blackholing vulnerable to maliciously-intended blackhole requests, i.e. *blackjack attacks*. BGP blackjack has been previously explored in [5] as a variation of the BGP hijacking (or prefix hijacking) attack [6].

A blackjack attack targets a specific prefix and attempts to trigger a blackhole for it without warning the legitimate owner of the prefix. This causes legitimate traffic to be dropped, essentially resulting in a DDoS attack. It can also be used to abuse route selection algorithms, which usually give preference to smaller routes [7] or routes tagged with a blackholing community [8] to re-route traffic. Despite being called an attack, blackjacks can also happen due to misconfigurations of an AS. These may not be necessarily malicious, but can still cause similar undesirable side-effects.

In this research, we aim to explore the importance of temporal representations of data for detecting blackjack attacks in BGP data. To achieve this, we use a supervised learning based approach and three techniques: (i) With timestamps (explicit representation) or without timestamps (implicit representation via packet sequence); (ii) A buffer (sliding window) of sequential (historical) data with/without timestamps; and (iii) A buffer of overlapping / non-overlapping data (via tap-delay-line) with/without timestamps. In order to evaluate the effects of these temporal representations, we train and test Decision Tree and Naive Bayes classifiers on a labelled blackhole and blackjack activity dataset which is generated using publicly available BGP data. To the best of our knowledge, this is the first work in the literature exploring the temporal representations of contextual data in the form of a sliding window, and the use of tap-delay-lines for detecting BGP blackjack attacks.

The rest of the paper is organized as follows. Section II summarizes the related literature on BGP blackholing and blackjack attacks. Section III describes the process of generating the blackhole and blackjack dataset as well as the credence metric for BGP data. Section IV introduces the propsoed approach used in this research. Section V presents the experiments and discusses the results. Finally, conclusions are drawn and future work is discussed in Section VI.

## II. Background and Related Work

BGP data is known to be very rich in terms of routing and AS related information. The analysis of BGP data could provide not only the topology and overall behaviour of ASes,

but also possible attempts at attacks. As a result, this could lead to better management, performance and security for a given organization. One of the attacks that can be performed is the BGP *prefix hijacking* [6]. A prefix gets hijacked when an AS announces it, even though it does not own it. This can cause multiple ASes to forward traffic intended for that prefix to the attacker, causing the packets to be lost and any service offered by the victim to be denied. BGP blackholing is a technique used by AS network managers for mitigating these types of DDoS attacks. It is a strategy of redirecting anomalous traffic to null interfaces. The most common implementation involves configuring the *next-hop* field of the attacked (victim) network to a private Internet address [9]. This, however, has the undesirable side-effect of rendering that particular network inaccessible, effectively contributing to the denial of the service.

Thus, an early improvement to this technique was developed in 2004 — RFC 3882 [10] — where by using BGP communities, the traffic would be blackholed (dropped) on individual border routers, as opposed to the entire AS. This strategy works by adding a BGP community that identifies each router on the announcement. If a router receives an announcement containing its community, it configures the blackhole for the announced addresses. Other routers would be oblivious to the blackhole request and would install the route normally (without a blackhole). In 2009, further improvements to this technique were proposed by using it in addition to *Unicast Reverse Path Forwarding* (uRPF) [11]. uRPF proposes that routers filter packets based on their source IPs as well as their destinations. By using uRPF, a router can determine if the route used is valid (or at least likely) for the provided source IP address. Packets from invalid (or unlikely) routes are dropped. More recently, in 2016, RFC 7999 [12] proposed the use of a well-known blackholing community in an attempt to streamline the implementation of blackholing-aware ASes and the relationship between them.

Among related works, Giotsas et al. [13] present the efficacy of BGP blackholing and discuss that blackholing is very useful in mitigating DDoS attacks. However, they do not have any analysis regarding the origin of the blackholing request, hence leaving open whether requests are legitimate or, possibly, malicious. Streibelt et al.'s [2] study, where the misuse of BGP communities is explored, reports that communities are meant to span only a limited number of peer ASes. But, in their datasets, communities were observed being forwarded far further than just a couple of hops. It is also reported that, in scenarios of misconfigurations, BGP communities can be used as an attack vector to cause malicious blackholing, route steering and manipulation (as in BGP blackjacks). Some of these attacks can be executed even without previously hijacking the target prefix.

The study on BGP blackholing in the wild by Jonker et al. [3] explores more technical aspects of the use of the blackholing technique. Their analysis on deployment of blackholes during DDoS attacks results in very relevant information on reaction time for deploying/withdrawing blackholes, attack duration, intensity and total number of packets. Furthermore, Meyer et al. [14] introduces an Intelligent Threat System for supporting in attack reconnaissance and mitigation. However, their use case for BGP blackholing does not account for possible malicious behaviours. Miller et al. [5] introduce the concept of blackjack attacks as the combination of a BGP blackhole request and a prefix hijacking attack [6]. They describe multiple possible variations of this attack as well as security mechanisms that could stop them. In this research, we concentrate on these BGP blackjack attacks.

## III. Dataset

Due to the unavailability of a labelled dataset for blackjack attacks, we compile a BGP traffic dataset based on the publicly available Route Views Archive Project (RVAP) [15]. The original data is around 178 million packets, May 5th 2020. Packets not identified as blackhole requests are discarded. For every blackhole request packet, one entry is created for each prefix announced. Each entry in the dataset is composed of a label classifying whether the packet is considered a benign blackhole request or a possible blackjack attack, along with the attributes shown in Table I. The process of filtering blackhole requests and labelling are discussed in the following sections. The pre-processed data used will be made public on Github.

### A. Filtering BGP Blackhole requests

A request for a BGP blackhole is made by sending a BGP announcement with the proper community attached to peer ASes. This community is established by each AS's administrator, so it can vary from AS to AS. However, since the addition of RFC 7999 [12], it has become more common for ASes to use the community suffix *:666*. However, it is naive to assume that all ASes follow this standard; either due to not having adopted it or due to security concerns [13].

In order to account for ASes that are not following the RFC 7999 standard, we compiled a dictionary, which is based on [13], of BGP communities that signal blackholing requests. A manual search on ISP webpages and AS registration documents is performed in order to augment this dictionary. The augmented dictionary contains *321* entries for *268* distinct ASes. This is used to determine whether a packet contains blackholing requests.

Thus, extracting BGP blackhole requests from the original data becomes a matter of checking if any of the communities in a given packet contain the *:666* prefix or are contained in the dictionary. If so, an entry is added to the dataset for each of the announced prefixes with a label 0 meaning "not an attack", and 1 meaning "possibly an attack". The following section presents the labelling process.

### B. Labelling of BGP Data

As seen in [5], there are multiple types of blackjack attacks. In this research, we focus on type *Type-0*, where the origin of the request is spoofed, and *Type-N*, where one or more peer connections are spoofed. To automate the process of labelling, an algorithm is designed to determine whether an entry is,

TABLE I
ATTRIBUTES IN THE BGP BLACKJACK DATA

| | |
|---|---|
| Time | The timestamp (in seconds) when the packet is received |
| Origin | The AS that originally created this request (the rightmost in the AS-PATH) |
| Number of Communities | Number of BGP communities attached to this packet |
| Size of AS-PATH | Number of hops in the AS-PATH property |
| Number of Prefixes | Number of prefixes announced by the packet |
| Number of Addresses | Number of addresses in the prefix range |
| Credence | Credence metric for the timestamp this packet received |

possibly, one of these two attacks. Essentially, for each entry in the dataset, two conditions are checked as reported in [4], [5], [7]:

1) Has this prefix been announced before? If so, is it being announced again by the same AS?
2) Does this blackhole request announce an AS-PATH that is more likely to be accepted than the existing one?

The first condition is to filter packets that announce prefixes not owned by the announcer. This is done by comparing their announcement to the previously seen announcements of those same prefixes. If more than one AS announces this prefix, there is a possible hijack attempt and being a blackhole request, the label is a *Type-0* blackjack. The second condition is to filter packets that try to exploit the route selection algorithm of the victim AS. It is common for ASes to accept routes that are shorter (less number of hops) than the ones installed [7]. Therefore, by announcing a prefix with a shorter AS-PATH, the attacker is more likely to make that route installed, which in return becomes a malicious blackhole.

As discussed earlier, BGP update messages are used for changing BGP routes. BGP routing changes could occur for many reasons, including hardware / software failures, routing policy changes, or malicious attacks. Determining the cause directly from the BGP update messages is almost impossible. Thus, network operators tend to allow route updates (announcements for prefixes) rather than run the risk of misrouting legitimate traffic. Hence, once a BGP update message with a previously unseen route is announced, it will likely be propagated to the AS's peers resulting in the number of packets being transmitted to increase. This would reflect as a spike in the BGP traffic where the packet load will be higher than it had been. In this work, we define the Credence Metric to measure the confidence in a given spike in BGP traffic.

Let $P$ be the set of packets received by a host. Let $time(p)$ where $p \in P$ be the timestamp when $p$ is received rounded to the nearest second. We start by calculating a time-series, $T$, where $T_i$ is the value of $T$ at instant $i$, as follows:

$$T_i = |[x | x \in P \wedge time(x) = i]| \tag{1}$$

Let $W$ be a number representing a time window under analysis such that $0 < W \leq |P|$. Next, we calculate a new time-series, $A$, where $A_i$ is the value of $A$ at instant $i$, as follows:

$$A_i = \frac{(T_{i-1} + ... + T_{i-W})}{W} \tag{2}$$

Time-series $A$ is only defined for $i > W$.

Let $H$ be a number indexing the historical data such that $W \leq H \leq |P|$. Let $hist(i)$, for instant $i$, be defined as:

$$hist(i) = [T_{i-1}, T_{i-2}, ..., T_{i-H}] \tag{3}$$

Time-series $hist$ is only defined for $i > H$.

Consider $max(X)$ as the maximum value for the set $X$. We now calculate a new time-series, $N$, where $N_i$ is the value of $N$ on instant $i$, with the noise metric at a given instant, as follows:

$$\begin{aligned} d_i &= |T_i - A_i| \\ m_i &= max(|x - A_i| \text{ for } x \in hist(i)) \\ N_i &= \frac{d_i}{d_i + m_i} \end{aligned} \tag{4}$$

Here, $d_i$ represents the difference between the number of packets at instant $i$ and the average number of packets in its immediate window. The value of $m_i$ represents the largest difference between any value in the history and the average number of packets in the immediate window of $i$.

With that, the Credence Metric is given relative to how much bigger the current 'difference' between the number of packets and the average number of packets is from the largest 'difference' seen in the immediate window of historical data. In summary, there are two parameters that can be customized in this approach: $W$ and $H$. In this work, $W$ is initialized as 12 and $H$ as 24, based on the empirical studies performed.

## IV. PROPOSED APPROACH

In this research, we explore the use of three different techniques for temporal representation of BGP data and evaluate how much each of them impacts the overall performance to detect BGP blackjack attacks. The first technique is the use of an explicit representation of time. In this context, an explicit representation of time is given in the form of a timestamp, i.e. the *time* attribute in the dataset. An implicit representation of time does not include that attribute. The second technique uses a sliding window of sequential (historical) BGP data over a given period of time. We refer to this as a buffer and use a buffer of size $X$. Buffers for sequential BGP packets in time with $X > 1$ will demonstrate overlapping data, i.e. oldest packet leaves from the buffer and the new one comes in. Thus, the third technique

TABLE II
EXPERIMENTS PERFORMED

| Experiment | Time | Buffer Size ($X$) | Tap-Delay-Line ($Y$) |
|---|---|---|---|
| 1 | Explicit | - | - |
| 2 | Implicit | - | - |
| 3 | Explicit | 5 | - |
| 4 | Explicit | 5 | 5 |
| 5 | Implicit | 5 | - |
| 6 | Implicit | 5 | 5 |
| 7 | Explicit | 7 | - |
| 8 | Explicit | 10 | - |
| 9 | Explicit | 12 | - |
| 10 | Explicit | 15 | - |

TABLE III
SIZE OF DATASET PARTITIONS IN EACH EXPERIMENT

| Experiment | Training (20%) | Testing (80%) |
|---|---|---|
| 1 | 20,792 | 503,701 |
| 2 | 19,804 | 463,637 |
| 3 | 41,512 | 1,056,575 |
| 4 | 8,962 | 240,332 |
| 5 | 41,404 | 1,050,858 |
| 6 | 8,944 | 238,119 |
| 7 | 43,984 | 1,146,835 |
| 8 | 45,966 | 1,238,851 |
| 9 | 46,760 | 1,282,000 |
| 10 | 47,610 | 1,325,242 |

introduces a tap-delay-line of size $Y$. As an example, instead of feeding each BGP packet sequentially ($Y = 1$) to an algorithm, we could feed every fifth ($Y = 5$) BGP packet. This reduces the data overlap ($1 < Y < X$) or completely avoids it ($Y = X$), which is analyzed in this work.

Given these techniques, we designed 10 experiments (Table II) to evaluate and analyze our proposed system. The size of the buffer is selected based on [16], which used a size of $X = 5$ as a starting point. We also explored the effect of different values for $X$ (5, 7, 10, 12 and 15). In all experiments, a balanced number of benign blackhole requests and blackjack attacks are used to train a CART Decision Tree classifier and a Naive Bayes classifier via the scikit-learn tool set. Naive Bayes and Decision Tree classifiers are used as the supervised learning algorithms, since (i) Naive Bayes is a simple probabilistic classifier; (ii) Decision Tree has the ability to select a subset of attributes from the set of all given attributes (using Information Gain metric); and (iii) Decision Tree solutions (trained models) are in the form of rules, hence making it human readable. These classifiers allow us to understand whether the data is linearly separable as well as compare the solutions. As seen in Table III, the training partition used is equivalent to 20% of the original dataset whereas the testing partition is equivalent to 80%. In all experiments, the number of blackjack attacks in the test partition represents around 0.5% of the total number of packets, similar to what is seen in real life [5].

## V. EXPERIMENTS AND RESULTS

In order to evaluate the performance of each experiment, we report the metrics of accuracy, Eq. 5 and recall for blackjack attacks, Eq. 6, as well as the solution complexity. Both equations are defined in terms of correctly classified entries as attacks (TP), correctly classified entries as normal (TN), incorrectly classified entries as attacks (FP), and incorrectly classified entries as normal (FN). In the following, we present the performance of the trained classifiers on the testing partition of the dataset. Further analysis of the attributes and solutions are done on the trained models.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

As seen in Fig. 1, the highest values for accuracy are reached in experiments 1, 3, 7 and 8 for the Naive Bayes classifier and experiments 3, 7, 8, 9 and 10 for the Decision Tree classifier; all higher than 80%. The common ones are experiments 3, 7 and 8 for both classifiers, where a sliding window of historical data including the timestamp attribute but no tap-delay-lines were employed as the temporal representation.

Even though the Naive Bayes classifier seems to be achieving a higher accuracy than the Decision Tree classifier in these experiments, a closer look shows that this is not the complete story. In Fig. 2, we observe that the Naive Bayes classifier performs very poorly (20% or less) in terms of recall. In other words, the Naive Bayes classifier is not able to learn to identify the blackjack attacks, which are only 0.5% of the test dataset. Thus, labelling the data as benign results in high accuracy but very low recall. On the other hand, the Decision Tree classifier reaches up to 85% recall while achieving 84% accuracy. This not only results in higher performance for the Decision Tree classifier but also shows that it learns to identify the blackjack attacks even though they only form the 0.5% of the dataset.

To put this in perspective, for example a dataset containing one million packets includes around 5,000 (0.5%) attack packets. Considering accuracy/recall upwards of 80%, the proposed system detects 4,000 of these packets without false alarms. Thus, for the remaining evaluations and analysis, we will focus on the solutions that uses the Decision Tree classifier.

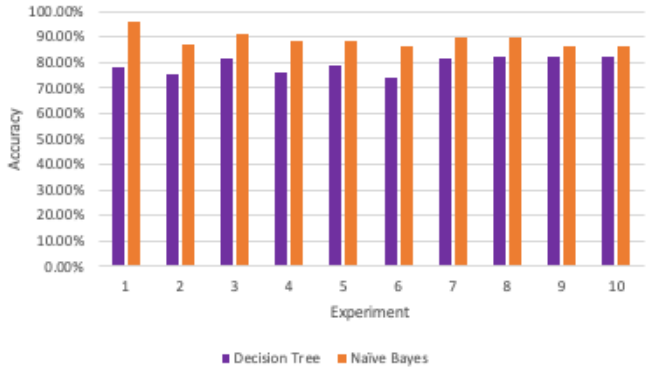


Fig. 1. Accuracy for each experiment

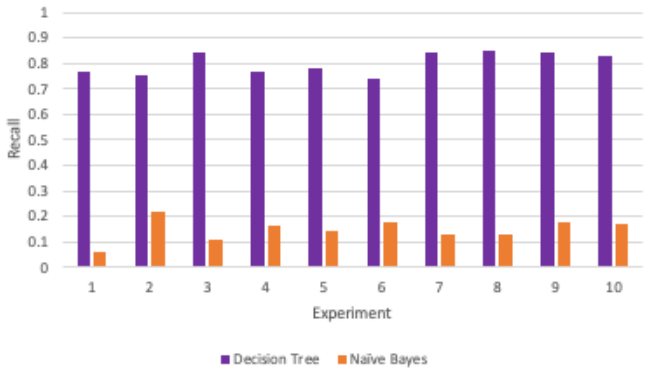


Fig. 2. Recall for each experiment

TABLE IV
TREE DEPTH OF THE DECISION TREE AFTER EACH EXPERIMENT

| Experiment | Tree Depth |
|---|---|
| 1 | 45 |
| 2 | 51 |
| 3 | 45 |
| 4 | 40 |
| 5 | 52 |
| 6 | 38 |
| 7 | 38 |
| 8 | 39 |
| 9 | 44 |
| 10 | 39 |

Table IV shows the depth of the trained decision tree after each experiment. The depth of the trained decision tree is used as a metric to indicate the complexity of the solution. Fig. 3 shows that the highest performance does not seem to require a complex solution. In fact, the trained solutions of the Decision Trees in experiments 7, 8 and 10 (which have upwards of 80% accuracy and recall) have some of the lowest complexities.

Furthermore, as seen in tables V to IX, we report how many times a specific attribute is used in the trained Decision Tree for each of the highest performing experiments: 3, 7, 8, 9 and 10. For experiments that use a buffer, the usage of each attribute is

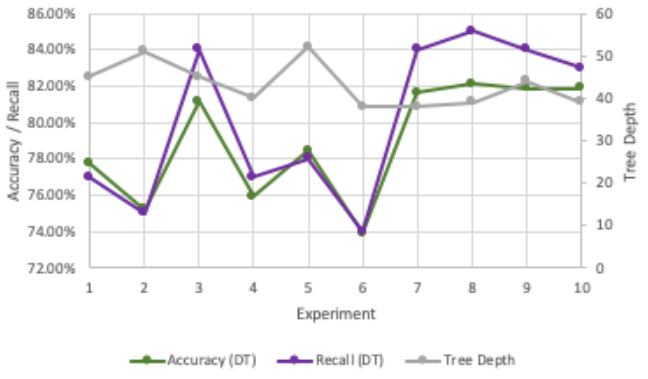


Fig. 3. Accuracy / Recall vs. tree depth for each experiment

TABLE V
NUMBER OF CONDITIONS FOR EACH ATTRIBUTE IN EXPERIMENT 3

| Attr. \Pkt. | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Origin | 694 | 498 | 518 | 504 | 584 |
| Num. Prefixes | 384 | 256 | 190 | 228 | 312 |
| Size AS-PATH | 334 | 306 | 278 | 294 | 336 |
| Time | 346 | 346 | 308 | 318 | 288 |
| Credence | 466 | 366 | 416 | 356 | 498 |
| Num. Communities | 26 | 4 | 10 | 12 | 18 |

TABLE VI
NUMBER OF CONDITIONS FOR EACH ATTRIBUTE IN EXPERIMENT 7

| Attr. \Pkt. | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Origin | 596 | 362 | 392 | 346 | 358 | 390 | 466 |
| Num. Prefixes | 294 | 186 | 170 | 184 | 170 | 174 | 240 |
| Size AS-PATH | 320 | 202 | 224 | 222 | 190 | 226 | 230 |
| Time | 220 | 178 | 178 | 220 | 192 | 176 | 174 |
| Credence | 364 | 248 | 212 | 234 | 214 | 266 | 290 |
| Num. Communities | 20 | 10 | 10 | 16 | 2 | 12 | 16 |

related to its packet in the buffer. For example, in Table V — detailing experiment 3 — the row for the *Origin* attribute and column 4 show that the *Origin* attribute relative to the fourth packet in the buffer is used 504 times in the trained decision tree. In these tables, cells with "-" indicate that they are not used (not applicable) for that experiment.

As stated before, experiments 3, 7, 8, 9 and 10 reached the highest recall performance for the Decision Tree classifier. What is interesting to notice here is that they all share the same set of temporal data representations: explicit representation of time, the use of a buffer (different sizes), and overlapping data, i.e. no tap-delay-line. Moreover, the results show that the use of tap-delay-lines seems to negatively impact the performance.

For the Decision Tree classifier, it is also interesting to notice that every experiment, with the exception of 4 and 6 (due to their use of tap-delay-line) performs better (see results above) than its counterpart. For example, we can see that experiment 1, which uses an explicit representation of time, performs better than experiment 2, which uses an implicit one. However, experiment 3, which uses a buffer in addition to an explicit representation of time, performs even better than experiment 1. We also observe that there is no correlation between the accuracy and recall vs. the solution complexity of the tree.

As for the attributes selected, the results show that the attribute *number of addresses* is not selected in any of the experiments by the decision tree. This seems to imply that this BGP attribute is not required for detecting blackjack attacks. This can be explained by the fact that most BGP blackholing requests are only accepted when they refer to a single IP (prefix /32), which would render that parameter to be 1 in the majority of the cases. Moreover, we also observe that, despite being used in all experiments, the attribute *number of communities* is used

TABLE VII
NUMBER OF CONDITIONS FOR EACH ATTRIBUTE IN EXPERIMENT 8

| **Attr. \Pkt.** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Origin | 510 | 360 | 258 | 284 | 258 | 248 | 228 | 272 | 264 | 344 |
| Num. Prefixes | 296 | 140 | 120 | 104 | 118 | 102 | 92 | 90 | 106 | 150 |
| Size AS-PATH | 238 | 166 | 156 | 144 | 126 | 164 | 150 | 150 | 146 | 146 |
| Time | 128 | 142 | 110 | 112 | 108 | 90 | 108 | 138 | 128 | 96 |
| Credence | 216 | 184 | 158 | 136 | 140 | 114 | 180 | 166 | 156 | 214 |
| Num. Communities | 24 | 2 | 6 | 6 | 12 | 6 | 16 | 16 | 10 | 16 |

TABLE VIII
NUMBER OF CONDITIONS FOR EACH ATTRIBUTE IN EXPERIMENT 9

| **Attr. \Pkt.** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Origin | 436 | 256 | 318 | 204 | 220 | 248 | 196 | 224 | 206 | 242 | 246 | 290 |
| Num. Prefixes | 260 | 148 | 132 | 78 | 100 | 86 | 66 | 88 | 90 | 84 | 100 | 112 |
| Size AS-PATH | 212 | 178 | 132 | 132 | 102 | 94 | 120 | 104 | 112 | 102 | 108 | 124 |
| Time | 68 | 88 | 92 | 96 | 104 | 86 | 82 | 78 | 90 | 78 | 94 | 86 |
| Credence | 186 | 148 | 150 | 118 | 128 | 114 | 106 | 94 | 124 | 116 | 132 | 188 |
| Num. Communities | 20 | 14 | 16 | 14 | 6 | 6 | 10 | 14 | 4 | 12 | 12 | 14 |

TABLE IX
NUMBER OF CONDITIONS FOR EACH ATTRIBUTE IN EXPERIMENT 10

| **Attr. \Pkt.** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Origin | 402 | 224 | 202 | 176 | 160 | 218 | 136 | 166 | 144 | 148 | 174 | 192 | 188 | 212 | 210 |
| Num. Prefixes | 222 | 108 | 96 | 60 | 82 | 74 | 96 | 64 | 56 | 64 | 62 | 66 | 70 | 74 | 86 |
| Size AS-PATH | 192 | 146 | 126 | 90 | 82 | 72 | 104 | 82 | 72 | 92 | 118 | 88 | 116 | 86 | 98 |
| Time | 70 | 88 | 72 | 74 | 78 | 46 | 58 | 66 | 72 | 60 | 70 | 68 | 54 | 74 | 66 |
| Credence | 132 | 108 | 108 | 112 | 112 | 112 | 72 | 76 | 88 | 90 | 90 | 78 | 108 | 130 | 138 |
| Num. Communities | 12 | 8 | 8 | 8 | 8 | 4 | 2 | 4 | 2 | 6 | 6 | 4 | 4 | 10 | 6 |

much less than the other attributes. This makes it a potential candidate for pruning of the decision tree. Finally, given the above results, we observe that experiment 8 — buffer size of 10 with explicit representation of time and overlapping data — achieves the highest recall (85%) and accuracy (84%) as well as the lowest solution complexity (depth 39) of the trained Decision Tree solutions.

## VI. CONCLUSIONS AND FUTURE WORK

Despite the benefits as a mitigation technique, BGP blackholing can also be used as an attack vector to cause service disruption, i.e. denial of service. This type of malicious behaviours are called BGP blackjack attacks. By using a BGP blackhole communities dictionary based on [13], we were able to filter publicly available BGP data for blackhole requests. Then, using the definition of the blackjack attacks [5], we were able to label possible *Type-0* and *Type-N* attacks using a credence metric. This allowed us to compile a BGP blackjack attack dataset. Using this publicly available compiled data, we explore three different temporal representations of data to detect blackjack attacks via supervised learning, namely Decision Tree and Naive Bayes classifiers. To this end, experiments are designed to evaluate the impact of temporal data representations: (i) With / without timestamps; (ii) with a buffer (sliding window) of sequential data with/without timestamps; and (iii) with a buffer of overlapping / non-overlapping data with/without timestamps. Results show that the solutions using a Decision Tree classifier outperform the ones using a Naive Bayes one. Moreover, temporal data representation in the form of a buffer of size 10 with time attribute and with overlapping data provides a good trade-off between accuracy, recall and solution complexity for the Decision Tree classifier. We also evaluate the relevance of the attributes chosen and select a subset for BGP Blackjack attack detection. To this end, 5 of the 7 BGP attributes — Time, Origin, Size-of-AS-Path, Number of Prefixes, and Credence — are observed to be the most relevant ones for this problem on our datasets. These observations and findings seem to be essential for the development of a high performing solution.

Future work will investigate the use of more datasets, variations on tap-delay-lines and other machine learning algorithms to improve the proposed approach. Also, online implementations of the proposed approach will be studied.

## ACKNOWLEDGEMENTS

The authors would like to thank Dr. Giotsas for providing a copy of the dictionary compiled in [13]. This research was supported in part by the Natural Science and Engineering Research Council of Canada. It is conducted as part of the Dalhousie NIMS Lab at: *https://projects.cs.dal.ca/projectx/*.

## REFERENCES

[1] Y. Rekhter, T. Li, and S. Hares, "A border gateway protocol 4 (bgp-4)," RFC 4271, January, Tech. Rep., 2006.

[2] F. Streibelt, F. Lichtblau, R. Beverly, A. Feldmann, C. Pelsser, G. Smaragdakis, and R. Bush, "Bgp communities: Even more worms in the routing can," in *Proceedings of the Internet Measurement Conference 2018*, ser. IMC '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 279–292. [Online]. Available: https://doi.org/10.1145/3278532.3278557

[3] M. Jonker, A. Pras, A. Dainotti, and A. Sperotto, "A first joint look at dos attacks and bgp blackholing in the wild," in *Proceedings of the Internet Measurement Conference 2018*, ser. IMC '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 457–463. [Online]. Available: https://doi.org/10.1145/3278532.3278571

[4] S. Murphy, "Bgp security vulnerabilities analysis," 2006.

[5] L. Miller and C. Pelsser, "A taxonomy of attacks using bgp blackholing," in *Computer Security – ESORICS 2019*, K. Sako, S. Schneider, and P. Y. A. Ryan, Eds. Cham: Springer International Publishing, 2019, pp. 107–127.

[6] X. Shi, Y. Xiang, Z. Wang, X. Yin, and J. Wu, "Detecting prefix hijackings in the internet with argus," in *Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC*, 11 2012, pp. 15–28.

[7] M. Caesar and J. Rexford, "Bgp routing policies in isp networks," *IEEE Network*, vol. 19, no. 6, pp. 5–11, 2005.

[8] "Cisco. remotely triggered black hole filtering - destination based and source based," 2005. [Online]. Available: https://www.cisco.com/c/dam/en/us/products/collateral/security/ios-network-foundation-protection-nfp/prod\textbackslash_white\textbackslash_paper0900aecd80313fac.pdf

[9] Y. Rekhter, R. G. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear, "Address allocation for private internets," Internet Requests for Comments, RFC Editor, BCP 5, February 1996, http://www.rfc-editor.org/rfc/rfc1918.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc1918.txt

[10] D. Turk, "Configuring bgp to block denial-of-service attacks," Internet Requests for Comments, RFC Editor, RFC 3882, September 2004.

[11] F. Baker and P. Savola, "Ingress filtering for multihomed networks," Internet Requests for Comments, RFC Editor, BCP 84, March 2004, http://www.rfc-editor.org/rfc/rfc3704.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3704.txt

[12] T. King, C. Dietzel, J. Snijders, G. Doering, and G. Hankins, "Blackhole community," Internet Requests for Comments, RFC Editor, RFC 7999, October 2016.

[13] V. Giotsas, G. Smaragdakis, C. Dietzel, P. Richter, A. Feldmann, and A. Berger, "Inferring bgp blackholing activity in the internet," in *Proceedings of the 2017 Internet Measurement Conference*, ser. IMC '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 1–14. [Online]. Available: https://doi.org/10.1145/3131365.3131379

[14] P. Meyer, R. Hiesgen, T. C. Schmidt, M. Nawrocki, and M. Wählisch, "Towards distributed threat intelligence in real-time," in *Proceedings of the SIGCOMM Posters and Demos*, ser. SIGCOMM Posters and Demos '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 76–78. [Online]. Available: https://doi.org/10.1145/3123878.3131992

[15] University of Oregon, "Route views project," https://www.routeviews.org/, [Online; accessed February 8th, 2020].

[16] P. Lichodzijewski, A. Nur Zincir-Heywood, and M. I. Heywood, "Host-based intrusion detection using self-organizing maps," in *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No.02CH37290)*, vol. 2, 2002, pp. 1714–1719 vol.2.