BGP hijacking classification

Shinyoung Cho
Stony Brook University
shicho@cs.stonybrook.edu

Romain Fontugne
IIJ Research Lab
romain@iij.ad.jp

Kenjiro Cho
IIJ Research Lab
kjc@iijlab.net

Alberto Dainotti
CAIDA, UC San Diego
alberto@caida.org

Phillipa Gill
UMass Amherst
phillipa@cs.umass.edu

***Abstract*—Recent reports show that BGP hijacking has increased substantially. BGP hijacking allows malicious ASes to obtain IP prefixes for spamming as well as intercepting or blackholing traffic. While systems to prevent hijacks are hard to deploy and require the cooperation of many other organizations, techniques to detect hijacks have been a popular area of study. In this paper, we classify detected hijack events in order to document BGP detectors output and understand the nature of reported events. We introduce four categories of BGP hijack: typos, prepending mistakes, origin changes, and forged AS paths. We leverage AS hegemony – *a measure of dependency in AS relationship* – to identify forged AS paths in a fast and efficient way. Besides, we utilize heuristic approaches to find common operators' mistakes such as typos and AS prepending mistakes. The proposed approach classifies our collected ground truth into four categories with 95.71% accuracy. We characterize publicly reported alarms (*e.g.* BGPMon) with our trained classifier and find 4%, 1%, and 2% of typos, prepend mistakes, and BGP hijacking with a forged AS path, respectively.**

I. Introduction

The Border Gateway Protocol (BGP) is the Internet's de facto inter-domain routing protocol [1]. It allows an Autonomous System (AS) to advertise the set of IP prefixes it manages as well as routes to destinations that its neighbors can reach by routing traffic towards it. BGP is based on trust, where an AS is supposed to announce only IP prefixes it owns and legitimate paths to destinations. However, malicious ASes can take advantage of this trust model by announcing others' IP prefixes or by forging AS paths [2]. These techniques are generally referred to as BGP hijacking [3]. BGP hijacks have been a problem on the Internet for over 20 years [4], with routing incidents regularly occurring. In 2018, 4,739 routing incidents have been disclosed by BGPmon, a popular monitoring service [5]. Once an AS hijacks a prefix, the AS can blackhole or intercept the hijacked traffic, or impersonate the legitimate receiver of the traffic [6]. Moreover, the hijacker AS can use the hijacked IP prefixes for spamming [7], [8].

While work on path validation [9] and RPKI [10] is actively underway in the IETF [11], deployment of these solutions that would prevent hijacking remains at an impasse. Since systems to prevent hijacks are difficult to deploy [12] and require the cooperation of many ASes, techniques to detect hijacks after they occur have been a popular area of study [13]–[26]. Existing works [13], [17] detect BGP hijacking by tracking whether any new pairs of neighboring ASes suddenly appear, or by searching for the violation of a BGP policy (*e.g.* a valley-free violation). However, the former approach may falsely capture enormous unrelated events, and the latter approach often relies on AS relationships that are difficult to infer accurately. Alternatively, ARTEMIS [25] accurately detects all attack configurations but only towards prefixes owned by the network running it, making it not applicable to detect attacks towards other prefixes or to global monitoring Besides, other BGP anomalies (*e.g.* link failure, misconfiguration) can make detection techniques less accurate or the interpretation of the detected event more difficult. More recently, some approaches aim to differentiate between different types of BGP anomalies, such as misconfiguration, link failure, or worm attacks [14], [15], [24], [27]–[29]. However, they do not aim to classify BGP hijack events. To our best knowledge, only Argus [13], [30] takes BGP hijacking into account in their classification of anomalies. Argus classifies BGP anomalies into four groups: link failure, hijacking, route migration, and traffic engineering. However, Argus only classifies hijacking that causes blackholes (not interceptions), and their system requires access to real-time data plane measurements to perform such classification.

In this work, we focus on distinguishing four types of BGP hijack events: *typos*, *prepending mistakes*, *origin changes*, and *forged AS paths*. We conjecture that these first two event types may be more indicative of a misconfiguration or human error, while the latter two may be more indicative of a malicious hijacking event. Note that route migration is not our focus in this paper. To classify events, we use supervised learning, a random forest (RF) classifier, with five features (Section IV-B). In our features, we leverage AS hegemony [31] to identify forged AS paths in a fast and efficient way without relying on inferred AS relationships. AS hegemony represents a score of dependency in AS relationships and we can obtain the scores of all ASes from IIJ's Internet Health Report (IHR) API [32]. For other classification features, we use heuristics that identify typos and prepending mistakes. To verify our approaches, we run our classifier on our collected ground-truth data that is manually verified, we then use the trained classifier to characterize candidate BGP hijacking events (*e.g.* BGPMon [33]). We expect that our classifier can help network operators to prioritize the handling of malicious BGP hijack attacks as opposed to the events by human error. In this paper, we make the following contributions:

Utilization of AS hegemony to detect forged AS path. We demonstrate that we can utilize AS hegemony to detect forged AS paths (Section IV-B). In addition, by using both definitions of AS hegemony (local and global AS hegemony) we improve

978-3-903176-17-1 / © 2019 IFIP

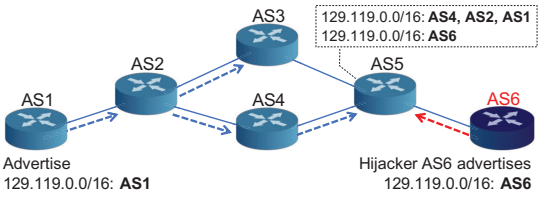

Fig. 1. EXAMPLE OF MULTIPLE ORIGIN ASES.

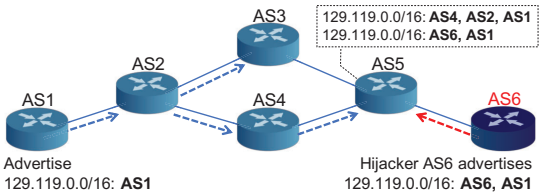

Fig. 2. EXAMPLE OF PATH MANIPULATION.

our detection accuracy.

Classifying different types of BGP hijacking events. We show that we can classify the four aforementioned BGP hijacking events: *typos*, *prepending mistakes*, *origin changes*, and *forged AS paths* using our features. Our approach carries over to other measurement databases such as those generated by BGPMon [33] and BGPStream [34]. With BGPMon's datasets, we find 4%, 1%, and 2% of typos, prepend mistakes, and BGP hijacking with a forged AS path, respectively.

II. BACKGROUND: ANOMALOUS BGP ANNOUNCEMENTS AND CAUSES

In this section, we overview the types of anomalous BGP events that we look for in traces of BGP control-plane messages. We first overview two suspicious events that may be observed in the control-plane and how they may be used by a malicious entity to perform prefix hijacking. We then overview two types of human error that may cause these types of anomalies. Our goal in this work is to distinguish cases of the first two anomalies that can be explained by the latter two sources of error.

Route origin change. A hijacker advertises to neighboring ASes a prefix that it does not own, illustrated in Figure 1. Since the BGP path selection process favors shorter paths (among paths where the next hop AS has the same LocalPref value or business relationship), other ASes may choose the illegitimate path announced by the hijacker if it is shorter than their paths to the legitimate AS. For example, AS5 may choose the illegitimate path announced by the hijacker (AS6) since the path is shorter.

In other cases, a hijacker announces a more specific version of the prefix announced by the legitimate AS. This type of BGP hijack is particularly problematic because routers using the longest prefix match will select and advertise this route. Thus, packets are forwarded to the hijacker rather than to the legitimate AS.

```
neighbor xx.xx.xx.xx route-map
test out
route-map test permit 10
set as-path prepend 47868 47868 47868 47868
```

Wrong prepending syntax: `set as-path prepend 47868 3`

Typo in prepending ASN: `set as-path prepend 47868 48768 47868 47868`

Fig. 3. EXAMPLE OF AS PATH PREPENDING.

AS-Path manipulation. Since an origin change or an origin legitimacy may be detected (*e.g.* via RPKI [10]), a hijacker may announce a forged path with its ASN on the path, but not as the route origin AS. The hijacker may place either the legitimate route origin AS or an unrelated AS as the route origin. For instance, Figure 2 shows that AS6 announces a fake path, [AS6, AS1], as if it is neighboring to the legitimate origin, AS1. By doing this, a hijacker can evade an origin authentication.

Typos in ASN or prefixes. When setting up routers, network operators have to type their ASNs and prefixes to the router configuration. In the process, there is a high chance for them to mistype ASN or prefixes. For instance, in May 2016, AS203959 announced prefix 191.86.129.0/24, which was a more specific prefix that another AS had announced. This incident was noticed because multiple origin ASes had announced the same prefix at the same time. Later on, it turned out that this was not an intended BGP hijack but just a typo. The reported hijacker, a network operator, mistyped the number 9 to 8 when typing its own prefix 191.96.129.0/24 [35].

Wrong AS path prepend. Another common error happens when network operators try to prepend their ASNs to announced paths. AS path prepending is a traffic engineering technique that consists in adding multiple times an ASN to a path so that the advertised path becomes less desirable due to its inflated path length. As shown in Figure 3, AS prepending mistakes can happen when an operator writes the number, "3", of repetitions of the ASN, "47868", instead of writing the ASN multiple times, "47868 47868 47868", or when an operator mistypes a sequence of the ASN such as "48768" instead of "47868". The former case results in an origin change and the latter case results in a forged AS path.

III. DATASETS

A challenge in a study of BGP hijacking is the limited number of sources of ground truth data - specifically, which events are intended (*i.e.* planned traffic engineering) versus those that are not (*i.e.* human error and potential hijacking attacks). In this section, we overview the ground truth data[1] we use to train our machine learning classifier as well as the longitudinal datasets we use to study the four routing event types described in Section II. A summary of the datasets that we use in our work is summarized in Table I.

A. Ground truth

We leverage two sources of unintended BGP routing events: potentially malicious hijacks and human error.

[1] https://github.com/grace71/bgp-hijacks-classifier

Potentially malicious hijacks. It is difficult to infer intent from BGP routing announcements. However, we can use the articles about the unexpected and occasionally impactful BGP events from the Dyn blog [36] as a source of routing announcements that are manually checked by experts and reported as suspicious.

Likely typos. As a second set of anomalous BGP events, we again use the articles on typos from the Dyn blog [35]. However, this type of report is rare so we also use data from the BGPMon [33] platform. BGPMon reports possible BGP hijacking events on a daily basis, and all reported events are the cases where an illegitimate AS announces a prefix or more specific prefix owned by another AS (e.g. multiple origin ASes). BGPMon reports only events that are highly possible to be BGP hijacking after excluding the obvious non-hijacking events. We use these events to find possible events caused by human error, for example, if the hijacked prefix (or ASN) is similar to the hijacker's legitimate prefixes (or ASN) and the routing announcement is withdrawn in a short time after the original announcement.

We also look for cases where the origin AS is a number, n, less than ten and when we observe a rapid withdrawal followed by the origin prepended n times. We consider such events as prepending mistakes by network operators (see Figure 2 for a specific example of this).

BGP data for ground truth events. After retrieving the details of each event from Dyn [36] and BGPMon [33], we collect historical BGP data using CAIDA's BGPStream [34], [37], which is an open source software framework for the analysis of both historical and real-time BGP data. To retrieve BGP data from CAIDA's BGPStream we need to know the prefix and likely time of the anomalous announcement/event. For events from Dyn we are able to retrieve relevant announcements for 35 events. Similarly, for BGPMon, we also retrieve the BGP data for 35 events, all labelled as typos and prepending mistakes (see Table I).

TABLE I
DATASET CHARACTERISTICS.

Dataset	Ground truth	BGPMon	BGPStream
Period	2008-02 ∼2018-07[a]	2018-05 ∼2019-02	2019-01-01 ∼2019-01-31
Total events	70	2,418	566
– w/MOAS	16	2,418	526
– w/New edges	18	-	40
– w/Typos	21	-	-
– w/Prepending	15	-	-
Avg # of paths[b]	669	795	157

[a]Our ground truth includes only the events reported by and manually checked, not all events that occurred during this period.
[b]The average number of AS paths per event.

B. Additional datasets

In addition to our ground truth data, we gather additional data from BGPMon [33] and BGPStream [34], [37] and characterize announcements and events in these datasets with the features described in the following section. While BGPMon gives a pre-filtered sample of highly likely BGP hijacks, BGPStream, in contrast, allows us to observe all anomalous BGP messages (*e.g.* multiple origin-AS prefixes) without filtering to perform our analysis. We perform our own filtering on data from BGPStream to avoid the cases that are likely traffic engineering (*e.g.* by avoiding incidents involving sibling ASes). We describe this in more detail in Section V-D

IV. METHODOLOGY

Our goal is to identify a set of features that can not only identify instances of multiple origin-AS prefix announcements and forged paths but also distinguish those instances from the cases likely caused by human error (*e.g.* typos or misconfiguring prepending). To accomplish this, we use machine learning to build a model of these events based on five main features using our ground truth dataset as input. A key metric used in two of our features is AS hegemony, which is a proxy for measuring the importance of an AS in the Internet graph [31]. These two features allow us to identify potential forged paths, which are difficult to infer accurately without knowing AS relationships. In this section, we first review the principles of AS hegemony then we introduce all the features we use for our classification.

A. AS hegemony

AS hegemony [31] is a metric that quantifies the likelihood of an AS to lie on paths toward certain destination IP prefixes. We distinguish two variants of this metric, *global* and *local* AS hegemony. The global AS hegemony is computed with paths to all IP prefixes globally reported by the BGP viewpoints. In this case, ASes with a large value stand for large transit networks that are commonly used to reach any host on the Internet. For instance, tier-1 ASes, like Level 3 (AS3356), have the highest scores, and stub ASes have the lowest scores.

The local AS hegemony is computed with paths from all BGP viewpoints towards only one origin AS. In this case, high values stand for ASes that are commonly used to reach the given origin AS. For instance, computing the local AS hegemony for UCSD (AS7377) reveals that the highest score is attributed to the Californian academic network, CENIC (AS2152), which is UCSD's main upstream provider. We obtain AS hegemony scores every 15 minutes for every AS using Internet Health Report (IHR) API [32].

B. Features

We now overview features we use to classify the different BGP events.

Number of valleys (Global hegemony) With the global AS hegemony, we identify BGP hijacking by path manipulation and the violation of BGP policy. Usually the global AS hegemony values corresponding to an AS path have only one local maximum around the middle of the path, which means large transit ASes are located in the middle of the AS path. However, if we find any valleys in global AS hegemony values, as illustrated in Figure 4, we define them as anomalies. In other words, if a rare transit AS with a low global AS

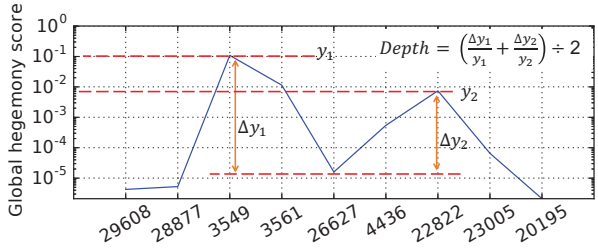

Fig. 4. A VALLEY ON AN HIJACKED AS PATH OF GLOBAL AS HEGEMONY. THIS EXAMPLE DRAWN FROM A DEMONSTRATION OF BGP INTERCEPTION AT DEFCON [38].

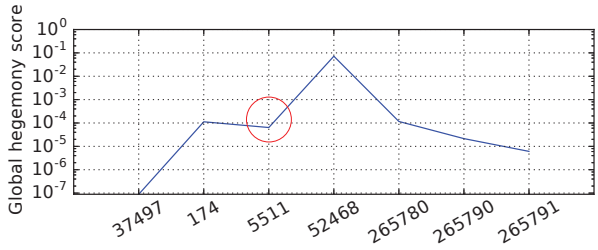

Fig. 5. EXAMPLE OF A SMALL VALLEY ON AN AS PATH.

hegemony score is located between two common transit ASes with higher scores, we define the dip between the two transit ASes as a valley. We consider the uncommon AS that causes the valley as the potential hijacker. For example, as illustrated in Figure 4, a hijacker, AS26627, is located between two tiers 1 ASes, CenturyLink(AS3561) and GTT(AS4436), and that is obviously suspicious [38].

In practice, due to approximation and measurement errors we observe many small valleys which may affect our classification. As illustrated in Figure 5, Orange (AS5511) generates a small valley on the AS path; however, Orange is not a hijacker, and Orange, Cogent (AS174), and UFINET (AS52468) are tier-1 or tier-2 networks. Thus, the small valleys should not be considered as an anomaly. To address this problem, we define how deep a minimum should be to be counted as a valley. Then, we ignore the negligible small valleys that are less than a certain threshold. We calculate the depth by taking an average of the rate of change between two local maxima, shown in Figure 4 for our ground truth data set. The calculated depth for each category is shown in Figure 6. We use 0.95 as

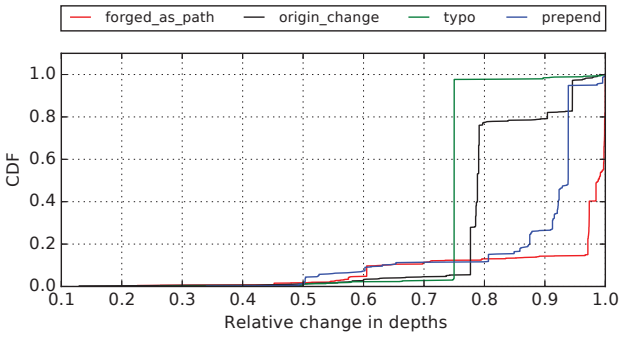

Fig. 6. CDF OF RELATIVE CHANGE IN DEPTHS ON EACH AS PATH IN OUR GROUND TRUTH DATA SET.

our threshold to count a dip as a valley. The threshold allows us to keep 85% of valleys in forged AS path, but to get rid of 99%, 99%, and 96% of valleys in origin change, typo, and prepend mistakes, respectively.

Similarity (Local hegemony) An AS that is closer to the origin AS on the AS path has a higher local AS hegemony score in general, and a higher score means that the AS is more important for the origin AS connectivity. If a hijacker creates a false AS paths or violates BGP policies, these changes may not be compliant with the previously computed local AS hegemony scores. As an example, all paths to AS14618 (Amazon) go through AS16509 (Amazon)[2] and thus a local hegemony of AS16509 for AS14618 is 1.0. If a hijacker announces a fake path between these two ASes, the announcement will cause a significant change on the local AS hegemony scores; thus the instances can be identified with this feature.

We use cosine similarity of the previous and current local hegemony as our feature. To get cosine similarity, we first recalculate local hegemony scores of origin AS during the potential BGP hijacking event. Next, we retrieve local hegemony scores of the same origin AS before the event through the IHR API [32]. Finally, we select the top three ASes in terms of local hegemony scores from each group and calculate the cosine similarity between them. This feature isolates ASes that rarely or never appeared on paths before but now become the major transit ASes for the origin AS.

Edit distance: Prefixes and Origin ASN. We use Levenshtein's edit-distance to identify typos made by network operators in route origin AS and prefixes. Our edit distance feature allows all possible edit operations, which are insertion, deletion, substitution, and transposition. For typos in origin ASN, we calculate the edit distance between the potential hijacker ASN and the victim ASN.

While detecting typos in origin AS is simple, identifying prefix typos is relatively complicated. To detect prefix typos, we first retrieve all prefixes of the potential hijacker's AS that are globally reported from BGP viewpoints before the BGP hijacking event occurred. Then, we calculate the edit distance between the hijacked prefix and each reported prefix; and return the minimum edit distance among them. Intuitively, operators are unlikely to make two mistakes in a single ASN or IP prefix; therefore, in both typo cases of origin AS and prefixes, if the edit distance is 1, it is highly likely to be a human error, not a hijacking.

Prepending mistakes To identify human error in AS path prepending (as shown in Figure 2), we use a simple method. For this type of error, the new origin AS will appear to be a small number (corresponding to the operator's desired amount of prepending) and the potential victim will appear as a direct upstream to this AS. So, we check whether the victim AS is a direct upstream of the new origin AS. Then, we assign the scaled probability against the number of origin ASN based on the prevalence of AS prepending observed on

[2]https://ihr.iijlab.net/ihr/14618/asn/?af=4&date=2015-03-26&last=7&hegemonydate=2015-03-26+07\%3A15&hegemonyy=y

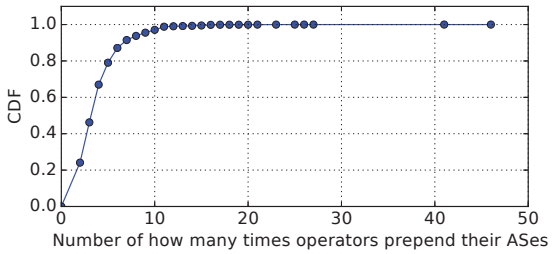

Fig. 7. CDF OF NUMBER OF AS PATH PREPENDING DURING 3.5 DAYS

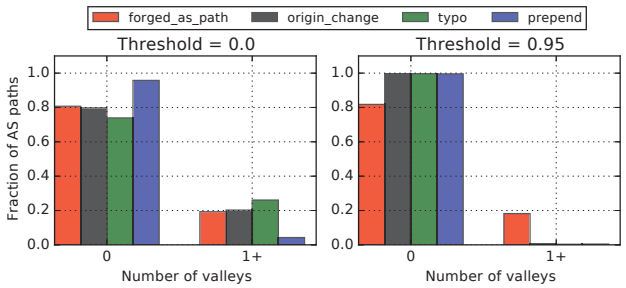

Fig. 8. NUMBER OF VALLEYS OF EACH AS PATH WITH DIFFERENT THRESHOLDS.

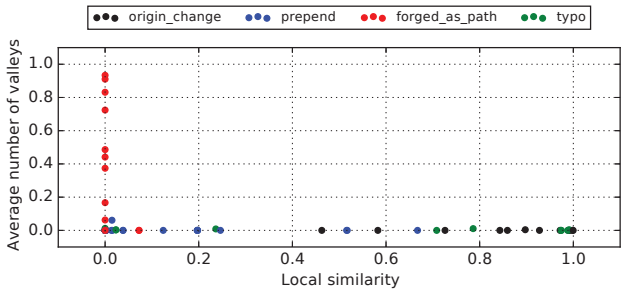

Fig. 9. COMPARISON BETWEEN AVERAGE NUMBER OF VALLEYS AND LOCAL SIMILARITY.

the Internet. Figure 7 shows the CDF of how many times ASes are prepended over a period of 3.5 days based on all AS paths to all IP prefixes, which are globally advertised on the Internet from BGP viewpoints. We use these results to compute the probability.

Multiple-origin AS (MOAS). We use a binary digit to indicate a MOAS conflict, which is one of the standard rules to determine BGP hijacking. These events may be potentially malicious hijacking or likely typos or errors as discussed in the prior sections, thus the presence of a MOAS violation cannot be the only feature to classify the events.

V. RESULTS

A. Evaluation of features

We evaluate (1) whether our features are able to distinguish the different types of events we have, and (2) the accuracy of our classifier.

Identifying forged AS path. As discussed earlier, because small valleys on AS paths add noise to our features, we set a threshold to filter out negligible small valleys. In order to understand whether the threshold removes the noise effectively, we analyze the number of valleys for each AS path with different thresholds. As illustrated in Figure 8, our results show that, with a threshold 0.95, most valleys on AS paths in human error are eliminated while the valleys of forged AS path remain.

From Figure 9, we get concern on the utilization of the feature because less than 20% of AS paths are identified with significant valleys. We compute the average number of valleys across all paths of each forged AS path event. As a result, we find that 53% of the events have an average number of valleys higher than 0.05. This indicates that the majority of these events contain at least a path with a valley. Thus, we can still utilize this feature to identify forged AS path events.

We now investigate forged AS path events that have a small average number of valleys to understand the reason. In our analysis, we observe a few hijackers placing themselves on a direct upstream position of a route origin AS. Because a hijacker is too close to the route origin AS, the hijacker does not create any valleys even when the hijacker has a low global hegemony score. We discuss this problem in details in Section VI. To mitigate this problem, we utilize local hegemony feature. Even if some hijackers yield small valleys, they can be detected with local similarity, illustrated in Figure 9. We find that 88% of samples in forged AS path cases have less than 0.002 in local similarity. Our manual inspection of the 12% events, where local similarity is larger than 0.002, reveals that in all these cases a hijacker announces an assigned but unused prefix using a forged AS path, which means no MOAS conflict is reported. Besides, the hijacker places its downstream customer as origin AS, so that local similarity is relatively larger than other cases.

Identifying human error. We now evaluate our features that are designed to identify human error. As illustrated in Figure 10, the prepending feature allows us to distinguish between forged AS path cases and prepending cases, although the local similarity values of these two types of events are similarly distributed. This observation also supports the complementarity of our features and the use of machine learning to combine all features and improve their discrimination power.

Our results of the edit distance feature are illustrated in Figure 11. The edit distance is derived from the minimum value of edit distances between a hijacked prefix and a set of prefix owned by a potential hijacker in our dataset. In

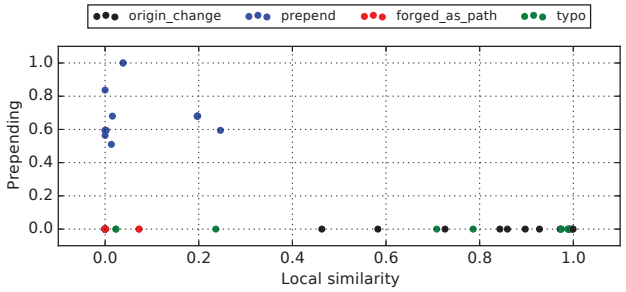

Fig. 10. COMPARISON PREPENDING FEATURE AND LOCAL SIMILARITY.

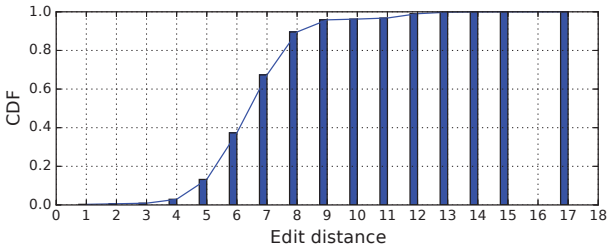

Fig. 11. EDIT DISTANCE BETWEEN A HIJACKED PREFIX AND A SET OF PREFIX OWNED BY A POTENTIAL HIJACKER.

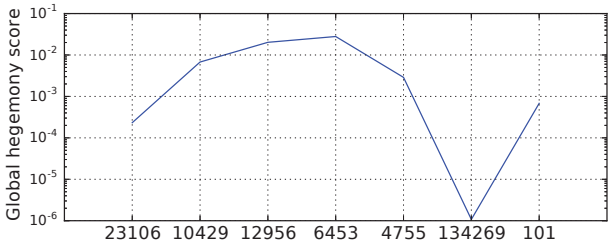

Fig. 12. EVENT OF FORGED AS PATH IN BGPMON.

Figure 11, we see that nearly 98% of the returned edit distance is larger than four.

B. The accuracy of our classifier

We use a random forest (RF) classifier with all our features to classify BGP hijacking events. To evaluate the accuracy of our classifier, we run the RF classifier on our ground truth dataset. First, we remove one event from the dataset. Then, we train our RF classifier with all other events and make the classifier predict the label for the removed event. We repeat this process for all events and count how many times the classifier has correctly classified the removed event. Using this experiment, we evaluate the accuracy of our RF classifier to **95.71%**. We also tried other classifiers, k-means and DBSCAN, and obtained the best results with RF. The accuracy of these classifiers is 54.5% and 61.9%, respectively. As some of our features are discrete, thus random forest (the use of decision tree) is the most appropriate.

C. Classification of BGPMon dataset

We train the RF classifier with our ground truth dataset, then predict classes of BGP hijacking events reported by BGPMon [33]. Figure 13 shows the result of our classifier for a total of 2418 hijacking alarms over ten months. We find 5%, 4%, 2% of forged AS path, typo, and prepend mistakes, respectively.

We further investigate and verify whether each event is classified correctly. As an example, in one of the forged AS path events, AS101 is pointed out as a hijacker by BGPMon against a prefix 103.100.12.0/24 owned by AS136650. However, we find that there is a forged AS path between a hijacker AS101 and AS134269, illustrated in 12. In this case, the valley created by the hijacker is clearly visible; furthermore the two networks are registered in two distance countries. AS101 stands for the University of Washington in USA and AS134269 is registered in India.

While investigating human error in AS path prepending in the BGPMon dataset, we find some unexpected cases. In these cases, both a potential hijacker and an original ASN are smaller than 10. We further investigate these cases with BGP measurements data over a period of two hours. We find that there are not only one but also more potential hijackers of which ASNs are all smaller than 10. Besides, the direct upstream AS of all the origin ASes is the same for all. For example, all route origins, ASN2 and ASN6, have only one direct upstream AS48420. We can infer that AS48420 tried to do AS path prepending with several numbers at that time, and BGPStream was confused with this and considered the repeated error as a change of an origin AS.

D. Classification of CAIDA's BGPStream dataset

Our approach carries over to another measurement dataset, CAIDA's BGPStream. BGPStream allows us to observe all anomalous BGP messages, and we focus on two events: multiple origin-AS prefixes and newly appeared pair of neighboring ASes. We perform our filtering on each event to avoid the cases that are likely traffic engineering (e.g., by avoiding incidents involving sibling ASes.) and to focus only on possible hijacking events. After running our trained classifier on events of multiple origin-AS prefixes, we find 1%, 1%, 14% of forged AS path, prepending mistakes, and typos, respectively from MOAS. For the events of newly appeared pair of neighboring ASes, we find 20%, 30% of forged AS path and typo, respectively.

VI. DISCUSSION

Limitation of features using AS hegemony. Our AS hegemony based features cannot classify forged AS paths if a hijacker is either tier-1 or tier-2 transit AS. These ASes already have a relatively high global hegemony, so they are less prone to create valleys on AS paths. Also, as many stub ASes generally depend on tier-1 or tier-2 transit ASes to transmit their traffic, a route origin AS of an AS path is likely to have a high dependency on those ASes. It means that we also might not find any anomalies on the AS path with local hegemony if the tier-1 or -2 networks are the upstream provider of the origin AS. However, in our ground truth, the real hijackers had relatively small hegemony, 0.00125 global hegemony on average. This is relatively smaller than 0.134 (Level 3) and 0.006 (UCSD).

Trial to fool edit distance. A hijacker may tailor an attack to make it similar to an operators mistake and fool our classifier. Due to the limited availability of IP address space, however, a hijacker has a limited number of choices to bypass and fool our detector. To explain this, we note that our edit distance features allow all possible edit operations, which are insertion, deletion, substitution, and transposition. A hijacker has to use only one operation among the set to avoid our detection. Computing the exact number of such possible edits is difficult. Thus, we check the number of the maximum possibilities of hijacking attacks where the number of edits is equal to 1. In other words,

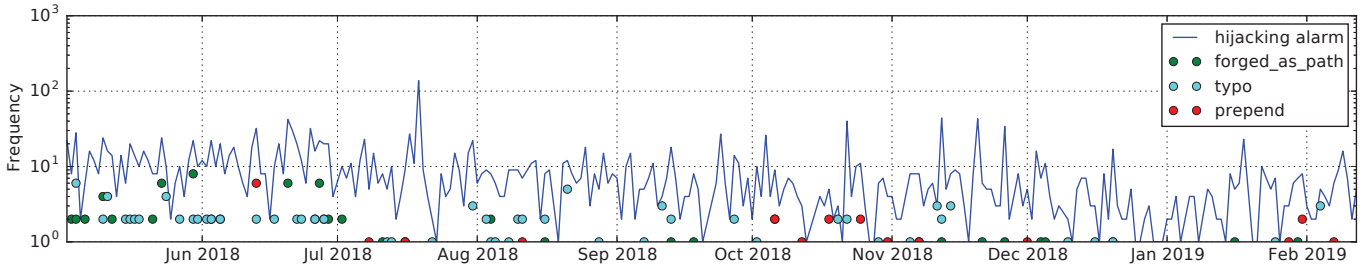

Fig. 13. CLASSIFICATION RESULTS OF BGPMON DATASET.

we look at the number of possibilities where the hijack can pretend to be a case of human error in the worst case scenario.

Assume that we have an IP address, 1.1.1.1/32. For insertion, a hijacker can insert any one digit from 0-9 before and after "1" in the first octet. The possibilities for each octet are 20 so that the total possibilities of insertion is 80. For deletion, because there are three numbers for each octet, the total possibilities of deletion are 12. For substitution, a hijacker can insert [1,2], [0-9], [0-5], respectively to each octet. Thus, the total possibility of substitution is 60. Transposition has the smallest number of possibilities, 8, because each octet is available for only two transposition operations.

Note that the total number of IP addresses is equal to 2^{32}. We consider the total number of possible edits and the number of possible cases with edit distance equal to 1. We find from the previous case that in the worst case scenario, the probability of fooling edit distance is negligibly small, which is equal to 3.73×10^{-8}.

VII. RELATED WORK

Detection of BGP Hijacking. To mitigate BGP hijacking, a number of detection techniques have been proposed [13]–[26]. Existing works detect BGP hijacking by searching for a MOAS violation or a BGP policy violation, or by tracking whether any new pairs of neighboring ASes suddenly appear. Argus [13] and Hu et al. [21] measure the number of hijacking attacks by correlating multiple sources of information from the data and control planes of the network and finding inconsistencies among them or checking reachability. Tahara et al. [22] use ping tests to detect similar attacks. Our work builds on these studies to detect BGP hijacks, but also take into account the possibility of human error.

A number of other studies also attempt to detect in real-time to protect the system from such attacks. PGBGP [17] designs heuristics to detect BGP hijacking and propose slowing down the propagation of such routes to allow human operators to respond to such attacks. Deshpande et al. [15] and Theodoridis et al. [16] use statistical analysis to identify anomalies and instabilities and thus detect BGP hijacks in real-time. Our work uses similar statistical techniques to detect BGP hijacks. However, unlike our work, these studies do not take into account the possibility of operator errors.

Classification of Internet Anomalies. Some works try to classify the different types of BGP anomalies observed in routing. The work [23] surveys the types of possible BGP anomalies (*e.g.* direct and indirect anomalies, link failure) and enumerates detection techniques with different approaches (*e.g.* machine learning, statistical pattern recognition). Marijana et al. [39] use standard statistical classification techniques to identify the types of BGP attacks, but do not consider human error. Some works [27]–[29] use machine learning techniques (*e.g.* SVM, HMMs) to classify BGP anomalies. However, these studies all focus on the classification between work attacks, link failure, and misconfiguration, and do not consider BGP hijacking. The study closest to our work is I-Seismograph [40]. It finds major changes in the Internet routing patterns and tries to identify the root cause behind it. However, unlike our work, it does not check if an error is by a hijacker AS or an operators mistake.

VIII. CONCLUSION

In this work, we leveraged AS hegemony and heuristic approaches to classify BGP hijack events into typos, prepending mistakes, origin changes, and forged AS paths. We improved the accuracy of our features by removing noise (e.g., small valleys) and using empirical measurements. In addition, our results show that even though each feature seems to be independent with each other, they can be leveraged together to enhance the accuracy of classification. Moreover, we use a random forest (RF) classifier with our five features and show that our classifier has 95.71% accuracy. We trained our classifier with our ground truth, and then used it to characterize candidate BGP hijacking events generated by BGPMon and BGPstream. With BGPMon, the results show that we classify 4%, 1%, and 2% of alarms as typos, prepend mistakes, and BGP hijacking with a forged AS path, respectively. With BGPStream, for multiple origin-AS prefixes events, we find that 1%, 1%, 14% of events are forged AS path, prepending mistakes, and typos, respectively. For events of newly appeared pair of neighboring ASes, we find that 20%, 30% of events have forged AS path and typo, respectively. Our results show that typos and prepending mistakes account for more number of events than origin changes and forged AS path.

For future work, we plan to utilize AS hegemony to localize a hijacker on an AS path. Localizing is possible by returning AS that is responsible for the valley on a path. In addition,

we are aware of the cases where we cannot detect a forged AS path with the number of valleys so we plan to look at different statistical techniques to mitigate such cases. Finally, in this paper, we focus on only BGP hijack attacks, but we plan to extend our datasets to include the data of all anomalies and utilize our features to characterize the anomalies. Also, we plan to increase the scope of human mistakes to cover configuration errors in routing policies.

ACKNOWLEDGEMENTS

We would like to thank the anonymous reviewers for their feedback. This work has been partially funded by the IIJ-II summer internship program. We acknowledge funding support from the NSF Grants CNS 1740895, CNS 1700657, CNS 1651784, CNS 1350720, CNS 1423659, CNS 1705024, OAC 1848641. This material is based on research sponsored by Air Force Research Laboratory under agreement number FA8750-18-2-0049. We also acknowledge partial support by the MSIT (Ministry of Science and ICT), Korea, under the "ICT Consilience Creative program" (IITP-2019-H8601-15-1011) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions in this paper are those of the authors and do not necessarily reflect the opinions of a sponsor, Air Force Research Laboratory, the United States, or the Republic of Korea Government.

REFERENCES

[1] Y. Rekhter, T. Li, and S. Hares, "RFC 4271: A border gateway protocol 4 (BGP-4)," Tech. Rep., 2006.
[2] S. Goldberg, M. Schapira, P. Hummon, and J. Rexford, "How secure are secure interdomain routing protocols," in *SIGCOMM*, 2010.
[3] H. Ballani, P. Francis, and X. Zhang, "A study of prefix hijacking and interception in the internet," in *SIGCOMM*, 2007.
[4] S. Misel, "Wow, AS7007! 1997."
[5] A. Robachevsky, "Routing security getting better, but no reason to rest!" https://www.manrs.org/2019/02/routing-security-getting-better-but-no-reason-to-rest/, 2019, online; accessed February 2019.
[6] O. Nordström and C. Dovrolis, "Beware of BGP attacks," in *SIGCOMM*, 2004.
[7] A. Ramachandran and N. Feamster, "Understanding the network-level behavior of spammers," in *SIGCOMM*, 2006.
[8] P.-A. Vervier, O. Thonnard, and M. Dacier, "Mind your blocks: On the stealthiness of malicious BGP hijacks." in *NDSS*, 2015.
[9] M. Lepinski and K. Sriram, "Bgpsec protocol specification," Tech. Rep., 2017.
[10] R. Bush and R. Austein, "The resource public key infrastructure (RPKI) to router protocol," Tech. Rep., 2013.
[11] IETF, "Secure interdomain routing (sidr) working group," http://datatracker.ietf.org/wg/sidr/charter/, online; accessed February 2019.
[12] P. Gill, M. Schapira, and S. Goldberg, "Let the market drive deployment: A strategy for transitioning to BGP security," in *SIGCOMM*, 2011.
[13] X. Shi, Y. Xiang, Z. Wang, X. Yin, and J. Wu, "Detecting prefix hijackings in the internet with argus," in *IMC*, 2012.
[14] Z. Zhang, Y. Zhang, Y. C. Hu, Z. M. Mao, and R. Bush, "iSPY: detecting IP prefix hijacking on my own," in *SIGCOMM*, 2008.
[15] S. Deshpande, M. Thottan, T. K. Ho, and B. Sikdar, "An online mechanism for BGP instability detection and analysis," *IEEE transactions on Computers*, vol. 58, no. 11, pp. 1470–1484, 2009.
[16] G. Theodoridis, O. Tsigkas, and D. Tzovaras, "A novel unsupervised method for securing BGP against routing hijacks," in *Computer and Information Sciences III*. Springer, 2013, pp. 21–29.
[17] J. Karlin, S. Forrest, and J. Rexford, "Pretty good BGP: Improving BGP by cautiously adopting routes," in *ICNP*, 2006.
[18] A. Haeberlen, I. C. Avramopoulos, J. Rexford, and P. Druschel, "NetReview: Detecting when interdomain routing goes wrong." in *NSDI*, vol. 2009, 2009.
[19] M. Lad, D. Massey, D. Pei, Y. Wu, B. Zhang, and L. Zhang, "PHAS: A prefix hijack alert system." in *USENIX Security symposium*, 2006.
[20] C. Zheng, L. Ji, D. Pei, J. Wang, and P. Francis, "A light-weight distributed scheme for detecting ip prefix hijacks in real-time," in *SIGCOMM*, 2007.
[21] X. Hu and Z. M. Mao, "Accurate real-time identification of ip prefix hijacking," in *Symposium on Security and Privacy (SP'07)*, 2007.
[22] M. Tahara, N. Tateishi, T. Oimatsu, and S. Majima, "A method to detect prefix hijacking by using ping tests," in *Asia-Pacific Network Operations and Management Symposium*, 2008.
[23] B. Al-Musawi, P. Branch, and G. Armitage, "BGP anomaly detection techniques: A survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 377–396, 2017.
[24] I. O. de Urbina Cazenave, E. Köşlük, and M. C. Ganiz, "An anomaly detection framework for BGP," in *2011 International Symposium on Innovations in Intelligent Systems and Applications*. IEEE, 2011, pp. 107–111.
[25] P. Sermpezis, V. Kotronis, P. Gigis, X. Dimitropoulos, D. Cicalese, A. King, and A. Dainotti, "ARTEMIS: Neutralizing BGP Hijacking within a Minute," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2471–2486, Dec 2018.
[26] Y.-J. Chi, R. Oliveira, and L. Zhang, "Cyclops: the AS-level connectivity observatory," in *SIGCOMM*, 2008.
[27] N. M. Al-Rousan and L. Trajković, "Machine learning models for classification of BGP anomalies," in *2012 IEEE 13th International Conference on High Performance Switching and Routing*, 2012.
[28] J. Li, D. Dou, Z. Wu, S. Kim, and V. Agarwal, "An internet routing forensics framework for discovering rules of abnormal BGP events," 2005.
[29] A. Lutu, M. Bagnulo, J. Cid-Sueiro, and O. Maennel, "Separating wheat from chaff: Winnowing unintended prefixes using machine learning," in *INFOCOM*, 2014.
[30] Y. Xiang, Z. Wang, X. Yin, and J. Wu, "Argus: An accurate and agile system to detecting ip prefix hijacking," in *ICNP*, 2011.
[31] R. Fontugne, A. Shah, and E. Aben, "The $thin$ bridges of AS connectivity: Measuring dependency using AS hegemony," in *PAM*, 2018.
[32] IIJ Research Lab, "Internet Health Report," https://ihr.iijlab.net/, 2019, online; accessed January 2019.
[33] BGPMon, "BGPStream," https://bgpstream.com/about/, 2019, online; accessed January 2019.
[34] CAIDA, "CAIDA BGP Stream," https://bgpstream.caida.org/, 2019, online; accessed January 2019.
[35] D. Madory, "BackConnects suspicious BGP hijacks," https://dyn.com/blog/backconnects-suspicious-bgp-hijacks/, 2016, online; accessed January 2019.
[36] Dyn, "Dyn blog," https://dyn.com/blog/category/research/, 2019, online; accessed January 2019.
[37] C. Orsini, A. King, D. Giordano, V. Giotsas, and A. Dainotti, "BGPStream: A software framework for live and historical BGP data analysis," in *IMC*. ACM, 2016.
[38] E. Zmijewski, "To catch a thief," https://dyn.com/blog/stealing-the-internet-back-1/, 2009, online; accessed January 2019.
[39] M. Ćosović, S. Obradović, and L. Trajković, "Classifying anomalous events in bgp datasets," in *CCECE*, 2016, pp. 1–4.
[40] M. Zhang, J. Li, and S. Brooks, "I-seismograph: Observing, measuring, and analyzing internet earthquakes," *IEEE/ACM Transactions on Networking (TON)*, vol. 25, no. 6, pp. 3411–3426, 2017.