Botgraph revised

capstone project proposal

team

|  |  |
| --- | --- |
| **Name** | **Registration Number** |
| Mohsin Ashraf | 15140094 |
| Mushood Hanif | 14140062 |
| Mehwish Hameed | 15140107 |
| Rida Yaqoob | 14140044 |

Table of Contents

[**Problem Statement** 3](#_Toc4450165)

[**Introduction** 3](#_Toc4450166)

[**Solution** 3](#_Toc4450167)

[Introduction 3](#_Toc4450168)

[Procedure 3](#_Toc4450169)

# **Problem Statement**

## **Introduction**

In 2016, Incapsula, A cloud-based application delivery platform generated a report regarding its internet traffic and found out that 51.8% of its traffic came from automated web bots including search engines, price scrappers, and viruses which cause **DDoS** attacks.These bots not only cause the leakage of business data, but also consume significant bandwidth and server overload.

**Bots** have been blamed for consuming a lot of internet traffic for some time now. Traditional bot detection studies focus on **signature-based** solutions but bots now forge identities to bypass detection through this method. There are **IP-based** solutions where each time a bot is detected the IP it corresponds to, gets blacklisted but that introduces an unnecessary element of maintaining a database of blacklisted IPs. Also, to bypass this, proxies and VPNs are easily available to mask and use an IP Address. Detection things like mouse movement and using JavaScript validation engines can be used but bots can now bypass all of this with the help of environments like **Headless Chrome.**

# **Solution**

## **Introduction**

Based on whether **authentication** is required or not, bots can be classified broadly into 2 categories: **social bots,** which target social networks and **web bots,** which target general websites. Compared to social bots, web bots are harder to detect because unlike social networks, there is no concept like the user account in the web traffic, and for a cloud provider, there must be millions of websites providing distinct services to its customers. Therefore, to efficiently identify bots among the websites, a generalized approach must be formulated for heterogenous scenarios.

## **Procedure**

BotGraph basically is performed in the following steps:

* Firstly, a **sitemap** is defined.
  + HTML **Request** parameters are
* Each user session is mapped to a subgraph of the sitemap which contains all the URLs the user has visited and their corresponding frequency of visiting.
  + \*get requests from server to map onto sitemap for subgraphs
* 2D images of the subgraphs based on the sitemap are generated to create a dataset on which our **model** will then train.
* A **CNN** boosted with evolutionary techniques with better performance than the regular state-of-the-art CNN will be employed with the task of learning the above dataset to classify any given sitemap into either “bot” or “non-bot”.
  + \*optimization techniques

# **SCOPE**

\*what kind of bots

\*what kind of sites

\*website used for testing

\*create our own dataset

\*writing crawlers (how many with purpose)

\*writing our own CNN