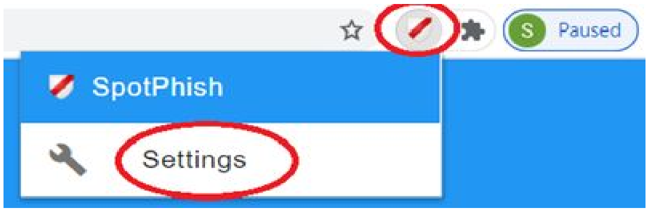
Modular Spotphish

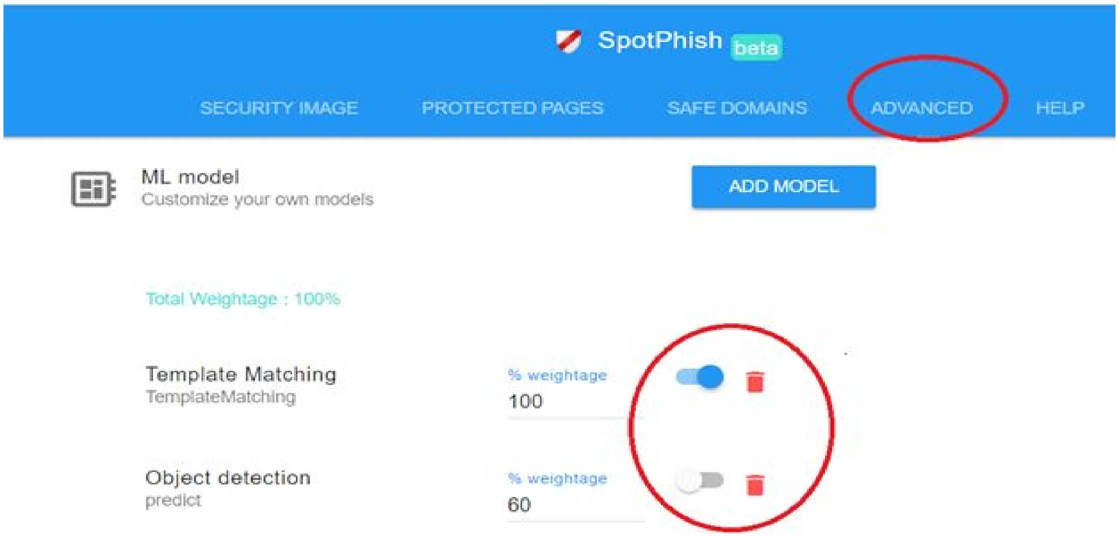
Spotphish has been made modular, with this release ML models can be picked and chosen on the fly. Models can be disabled, enabled and new models can be added online.

1. Enable/disable model

- Go to Settings



- Go to ADVANCED



ML model
Customize your own models

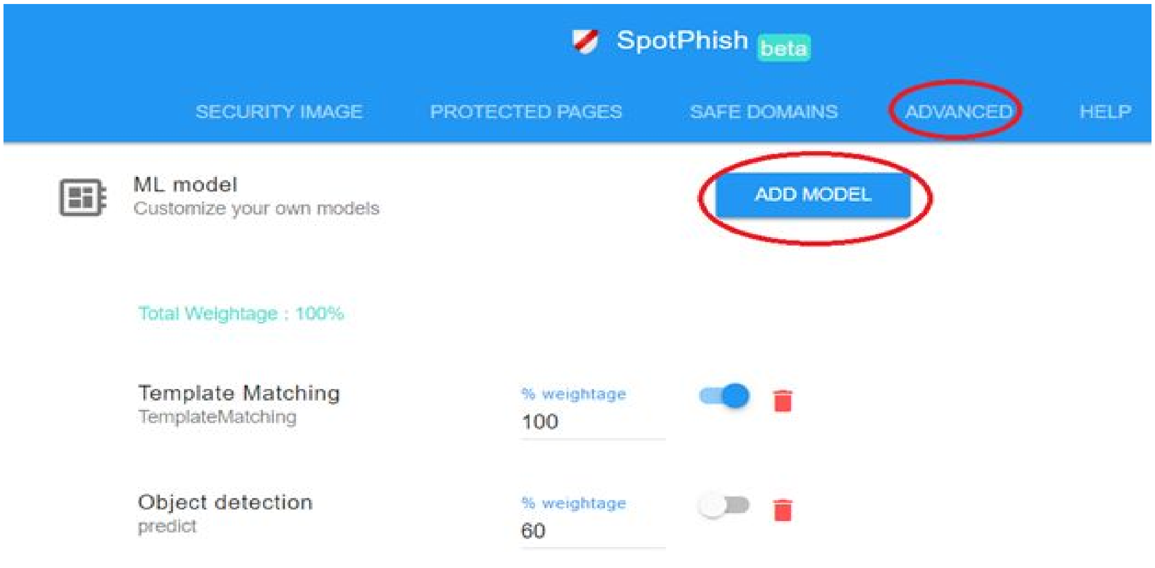
Total Weightage : 100%

Template Matching
TemplateMatching
% weightage
100

Object detection
predict
% weightage
60

2. Add new model

- Click ADD MODEL button



SpotPhish beta

SECURITY IMAGE PROTECTED PAGES SAFE DOMAINS ADVANCED HELP

ML model
Customize your own models

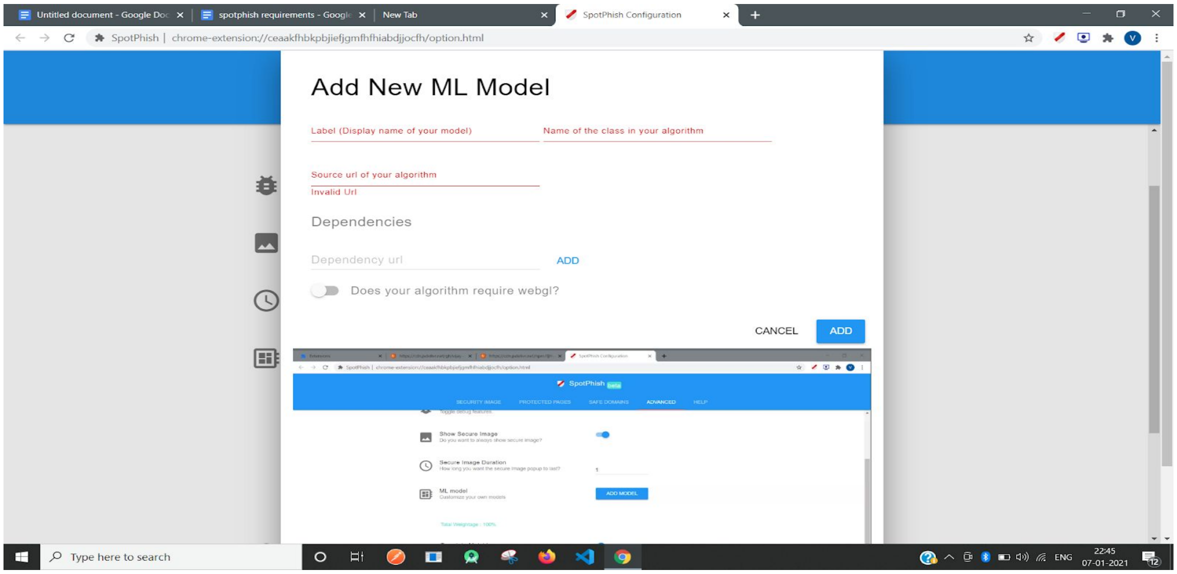
ADD MODEL

Total Weightage : 100%

Template Matching
TemplateMatching
% weightage
100

Object detection
predict
% weightage
60

- Popup for add model



Untitled document - Google Doc | spotphish requirements - Google | New Tab | SpotPhish Configuration

SpotPhish | chrome-extension://ceaakfhbkpbjiefjgmfhfhiabdjjocfh/option.html

Add New ML Model

Label (Display name of your model)

Name of the class in your algorithm

Source url of your algorithm
Invalid Url

Dependencies

Dependency url ADD

Does your algorithm require webgl?

CANCEL ADD

Type here to search

ENG 22:45 07-01-2021

Fields:

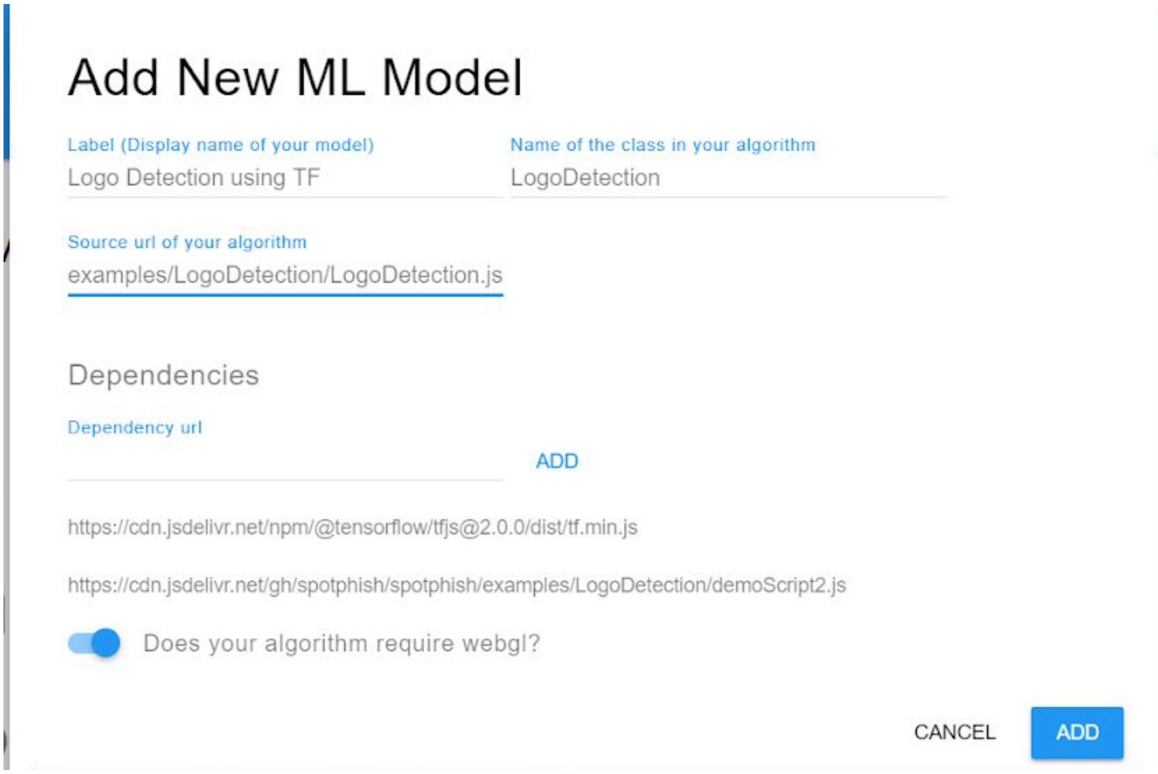
➢ Label : Any name with which you want to reference your model **(Required)**

- Name : Name of the class exported from your algorithm. This name must be unique. **(Required)**
- Source : Url of the algorithm where it is hosted. **(Required)**
  - **Note:** For now, only **cdn.jsdelivr.net** url are allowed. If your algorithm is hosted on github, then follow these steps to get CDN url.

- Dependencies : Your algorithm may depend on many other JS files, libraries or modules.
  - If you use your dependent files in the algorithm using **<script>** tag. Get the CDN url of each file and add those urls under dependencies.
  - If you use your dependent files in the algorithm using **import or require** then no need to add under dependencies. Instead replace those local urls in import() to CDN urls.
  - Please check all local paths are replaced by CDN url

- Webgl Switch : If your algorithm must run on GPU, then prevent it from running on CPU by switching on this field.

Note : A video is attached below the popup for your reference.



Add New ML Model

Label (Display name of your model)
Logo Detection using TF

Name of the class in your algorithm
LogoDetection

Source url of your algorithm
examples/LogoDetection/LogoDetection.js

Dependencies

Dependency url

https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@2.0.0/dist/tf.min.js

https://cdn.jsdelivr.net/gh/spotphish/spotphish/examples/LogoDetection/demoScript2.js

Does your algorithm require webgl?

- Click Add, ML model is added in the list

3. Assign weightage to models

- Total model weights should be 100%
- Distribute total weights (100%) to the enabled models

4. Train models

- Train models - it can be template matching, classification, object detection or any other model. Models can be trained using any algorithm and any tool.

- Trained models should accompany a javascript file(say main.js) which will have a predict function. Here is the structure of the JS file.

    a. A class should be exported
    b. Class must have a predict function.
    Input : src of img for which model will predict
    Output: json format output

    ```
    {
            class: <name of predicted class> (string),
            confidence: <confidence %> (int),
            time_taken: <time taken to predict> (int),
            image: <base64 url of predicted image>
    };
    ```

- main.js may have either imported other js files or may be using other js files by injecting them through <script> tag in some html file.
    a. If you use your dependent files in the algorithm using **<script>** tag. Get the CDN url of each file. You will need these later.
    b. If you use your dependent files in the algorithm using **import or require,** then replace those local urls to CDN urls in import().
    c. Please check all local paths are replaced by CDN url.

5. Advance users

- Go to **examples/LogoDetection** folder of https://github.com/spotphish/spotphish
    - LogoDetection.js is the main file which contains the algorithm of detecting a logo.
    - It depends on demoScript1.js (used by importing)
    - It depends on demoScript2.js (used by injecting script tag in index.html)
    - It depends on tensorflow library (used by injecting script tag in index.html)

- Converting github url to CDN url

  https://github.com/spotphish/spotphish/blob/master/examples/LogoDetection/LogoDetection.js

  Converted to

  https://cdn.jsdelivr.net/gh/spotphish/spotphish/examples/LogoDetection/LogoDetection.js

- Replacing file reference by CDN

  Import {sayHello} from “./demoScript1.js”;

  Replace to

  import {sayHello} from
  "https://cdn.jsdelivr.net/gh/spotphish/spotphish/examples/LogoDetection/demoScript1.js"

- Inside LogoDetection.js

```
import ...
export default class LogoDetection {
        predict(src) {
                img.src=src
                loadMyModel(https://cdn.jsdelivr.net/gh/spotphish/spotphish/examples/LogoDetection/model/model.json)
                runMyModel(img)
                return {
                        site: 'XYZ',
                        confidence: 0,
                        time_taken: 0,
                        image: base64 url of predicted image
```

```
                    }
          }
}
```

- Inside index.html

  If your dependent library is not delivered through cdn then download it,push it to your own github repo and get cdn url following the below steps.

  For example
  <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@2.0.0/dist/tf.min.js"></script>
  <script
  src="https://cdn.jsdelivr.net/gh/spotphish/spotphish/examples/LogoDetection/demoScript2.js"></script>

  Whatever is in script tags goes into the dependencies field of Add New ML Model form.

6. Helper tips

- Getting CDN url from github raw file
  - Get the link on github and click on “Raw” version.
  - Change raw.githubusercontent.com to cdn.jsdelivr.net
  - Insert /gh/ before <username>
  - Remove the branch name.
  - (Optional) Insert the version you want to link to, as @version (if you do not do this, you will get the latest - which may cause long-term caching)

  Examples:
  - http://raw.githubusercontent.com/<username>/<repo>/<branch>/path/to/file.js
    - Latest version
      http://cdn.jsdelivr.net/gh/<username>/<repo>/path/to/file.js

    - Specific version or commit hash
      http://cdn.jsdelivr.net/gh/<username>/<repo>@<version or hash>/path/to/file.js