Malware classification with spark

R08943095 張惟筑
R08921a17 賴正儒
R09921a18 王子元

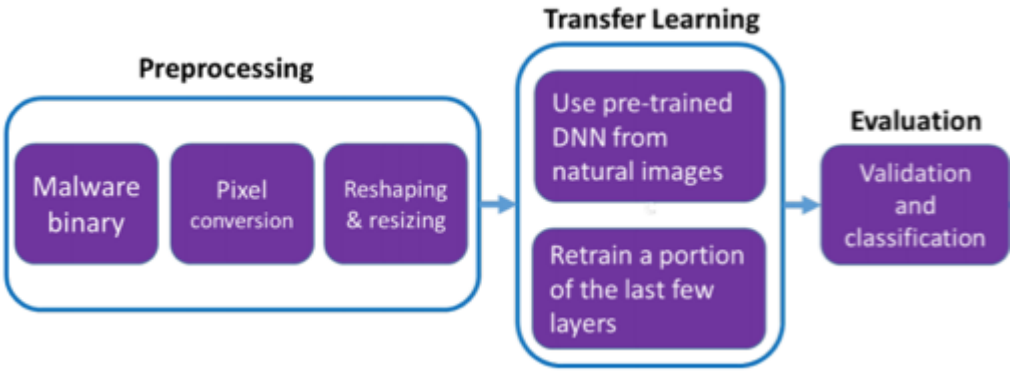
1. Abstract

With the progress of technology and the fast development of internet, various types of viruses emerge and spread rapidly on the internet. Therefore, to classify the source of viruses becomes more and more important. To reach our goal, we use transfer learning and train our classification model on the spark platform. After that, we build a web application with flask and deploy it to Amazon Web service (AWS) and simultaneously collect the usage data.

2. Approach

Github: https://github.com/jeremywangyuan/cloudcomp/tree/main/final
Youtube: **https://youtu.be/R4wIu6YfZEs**

- **Malware Classification flow**



For our malware classification problem, our goal is to classify input execution file (.byte) into labeled datas. We break it down to three parts:

(1) Background:
Malware classification is a competition on Kaggle hosted by MIcrosoft. We use data on Kaggle as our training and testing data.

(2) Preprocessing
Transform the input binary execution file into image and reshape the image to desired size.

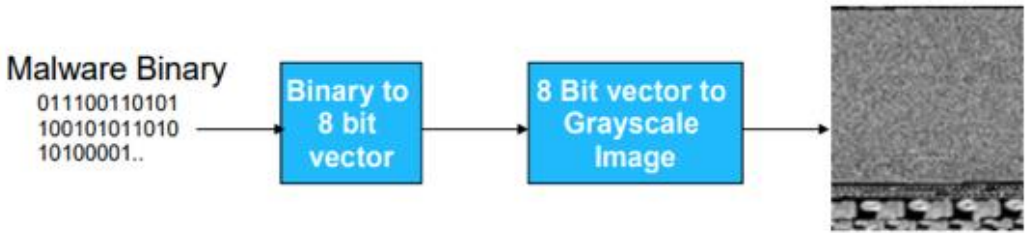


Fig.1 Visualizing Malware as an Image

(3) Transfer Learning

We use inception model as our backbone and do transfer learning.

(4) Evaluation

Use cross entropy loss to evaluate the accuracy of our model

- **Training on PySpark**

(1) Environment Setup

Due to some budget issues, we train our model with local PySpark environments.

```
#Import PySpark libraries
from pyspark.sql import SparkSession
from pyspark import SparkContext, SparkConf
from pyspark.sql import SQLContext

spark = SparkSession.builder.appName('Spark').config("spark.driver.memory", "16g").config("spar
spark
```

SparkSession - in-memory
SparkContext

Spark UI
Version
`v2.4.0`
Master
`local[*]`
AppName
`Spark`

(2) Training model

```
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
from pyspark.ml.classification import LogisticRegression
from pyspark.ml import Pipeline
from sparkdl import DeepImageFeaturizer

# model: InceptionV3
# extracting feature from images
featurizer = DeepImageFeaturizer(inputCol="image", outputCol="features", modelName="InceptionV3")

# used as a multi class classifier
lr = LogisticRegression(maxIter=10, regParam=0.03, elasticNetParam=0.5, labelCol="label")

# define a pipeline model
sparkdn = Pipeline(stages=[featurizer, lr])
spark_model = sparkdn.fit(train)
```

(3) Future work:

It can be deployed to cloud platform and upload the newest (best) model to our

cloud storage which can be fetched by our application.

- **Web Application**

(1) Overall Flow

(2) Front-end

We implement our application frontend with simple html and javascript. It can be migrated to modern frontend framework (VueJS, ReactJS) easily.

(3) Backend

a. Framework

Our backend server is implemented with python Flask framework.

b. Classification

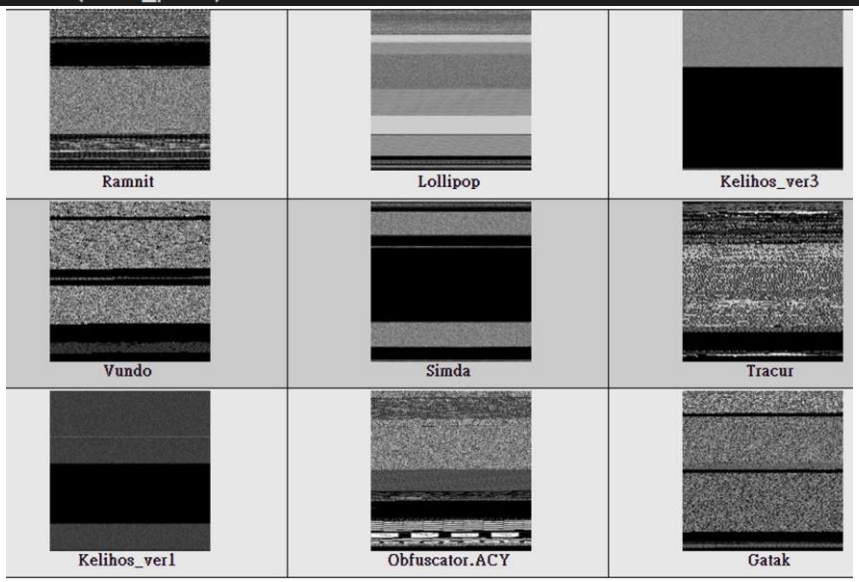
We load the model trained from Spark to do classification tasks.

```
# Define a flask app
app = Flask(__name__)

# Load trained model
MODEL_PATH = 'models/model.h5'

model = load_model(MODEL_PATH)

malware_class = {1: "Ramnit", 2: "Lollipop", 3: "Kelihos_ver3", 4: '

print('Model loading...')

print('Model loaded. Started serving...')

print('Model loaded. Check http://127.0.0.1:5000/')
```

c. Image Preprocessing

Transform input binary file into image format

```
w = 1024
h = os.stat(file_path).st_size//w
with open(file_path, 'rb') as img_file:
    byte = np.fromfile(img_file, dtype=np.int8, count=w*h).reshape(w, h)
f = Image.fromarray(byte).convert('RGB')
file_path = os.path.splitext(file_path)[0]+'.jpg'
f.save(file_path)
```



d. Data collection

It is important to know that the distribution of malware program in real world. We store the image file (after processing) to s3 bucket.

```
REGION = 'REGION'
BUCKET = 'BUCKET'
s3_connection = boto.connect_s3('ACCESSKEY', 'ACCESSVALUE', host='s3.us-east-2.amazonaws.com')
bucket = s3_connection.get_bucket(BUCKET)
key = boto.s3.key.Key(bucket, file_path)
key.set_contents_from_filename(file_path,
        cb=percent_cb, num_cb=10, policy='public-read')
```

e. Security issue

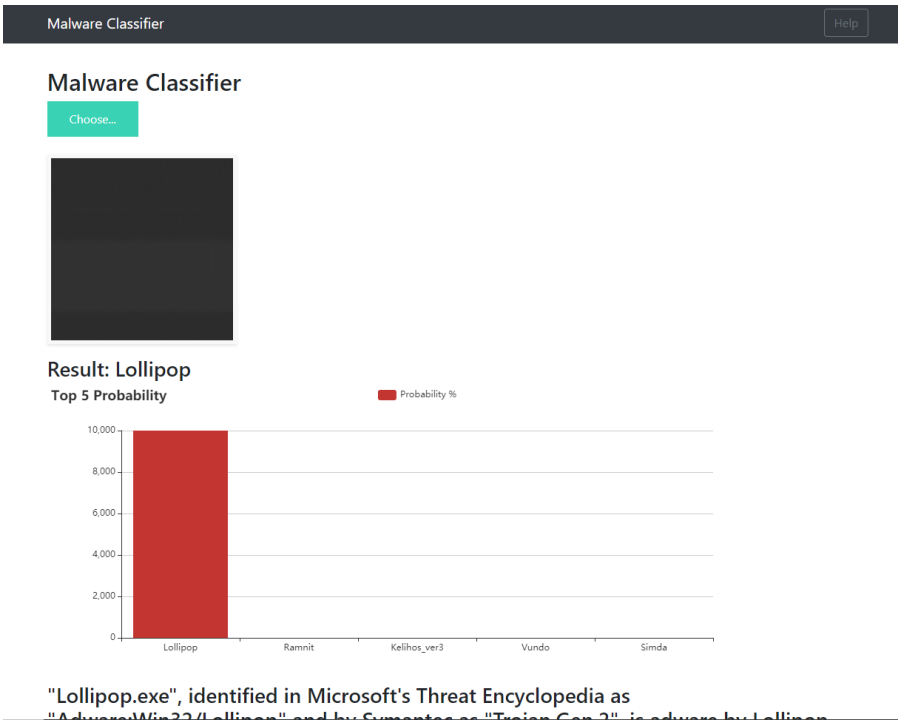
To protect our server, we need to check if the uploaded file format is correct

```
file_path = os.path.join(
    basepath, 'uploads', secure_filename(f.filename))
```

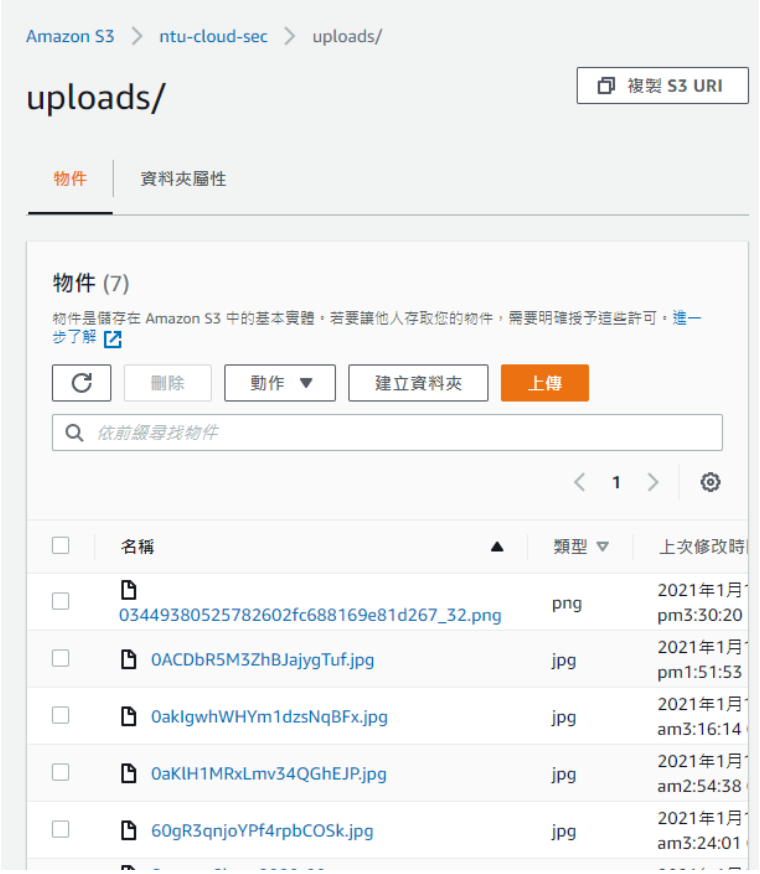
3. Results

(1) Init UI

(2) The result of classification



(3) S3 bucket



4. Discussion

(1) Hard to deploy our service to cloud platform

We had tried Microsoft Azure to deploy our application and design flow. However, due to some limitation and flexibility problem, we finally decided to use

AWS.

AWS contains the most kinds of service compared to other cloud platform. At first, we tried to use AWS Elastic Beanstalk to deploy our service, but still encounter some package problem. We found that we need to install some special package (eg. tensorflow without cache …etc) to deploy our service on AWS EC2. Therefore, we simply host an EC2 instance and install packages one by one. Finally, it works.

(2) Expensive to use cloud service

We tried to apply more fancy and automatic flow to our service. We found that we need some budgets to use PySpark on cloud platform. Instead, we train the model in our local environment.

(3) Not to upload the access key to github

To prevent malicious access to our AWS account, check it again that the secret key cannot be uploaded to GitHub.

5. Reference

(1) Microsoft Malware Classification Challenge (BIG 2015) https://www.kaggle.com/c/malware-classification/overview

(2) Tensorflow https://github.com/tensorflow/tensorflow

(3) Flask Documentation https://flask.palletsprojects.com/en/1.1.x/

(4) Python boto documentation https://boto3.amazonaws.com/v1/documentation/api/latest/index.html

(5) Transfer learning https://en.wikipedia.org/wiki/Transfer_learning