Manual & Security Audit for Train

ISIC2019II

Karl Kindermann

PADME

PADME

PADME-ANALYTICS.DE

First printing, January 31, 2024

Contents

1. Train Summary

1.1 Overview

- **Train name:** ISIC2019II
- **Creator of the train:** Karl Kindermann
- **Location:** `https://git.rwth-aachen.de/padme-development/padme-train-depot/-/archive/main/padme-train-depot-main.zip?path=ISIC2019II`

1.2 Image Building File

Listing 1.1: Image building file

```
FROM pytorch/pytorch:2.0.1-cuda11.7-cudnn8-runtime
RUN apt-get update && apt-get install -y git
WORKDIR /usr/src/app

LABEL "envs"="[{\"name\":\"FHIR_SERVER\",\"type\":\"string
   \",\"required\":true},{\"name\":\"FHIR_PORT\",\"type\":\"
   number\",\"required\":true},{\"name\":\"BATCH_SIZE\",\"type
   \":\"number\",\"required\":true},{\"name\":\"LR\",\"type
   \":\"string\",\"required\":true},{\"name\":\"WEIGTH_DECAY
   \",\"type\":\"number\",\"required\":true}]"

COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt

COPY . .

CMD [ "python", "./main.py" ]
```

1.3 Requirements & Dependencies

Listing 1.2: List of all Requirements

```
requests
Pillow
pandas
numpy
git+https://github.com/beda-software/fhir-py.git
torchvision
datetime
pytz
tqdm
pytest-shutil
```

1.4 Code Base

Legend: env query retrieve_previous_results execute_analysis save log

```python
from fhirpy import SyncFHIRClient
import requests
from io import BytesIO
import numpy as np
import pandas as pd
from PIL import Image
from torch.utils.data import Dataset
import torchvision.transforms as transforms

class ISICDataset(Dataset):
    classes = {'NV': 0, 'MEL': 1, 'BKL': 2, 'DF': 3, 'SCC': 4, 'BCC': 5, 'VASC': 6, 'AK': 7}

    """ISIC dataset."""
    def __init__(self, fhir_server, fhir_port, split='train', input_size=256):
        """
        Args:
            fhir_server (string): Address of FHIR Server.
            fhir_port (string): Port of FHIR Server.
        """
        self.fhir_server = fhir_server
        self.fhir_port = fhir_port
        self.split = split
        self.input_size = input_size
        # Create an instance
        client = SyncFHIRClient('http://{}:{}/fhir'.format(fhir_server, fhir_port))
        # Search for patients
        patients = client.resources('Patient')  # Return lazy search set
        patients_data = []
        for patient in patients:
            patient_birthDate = None
            try:
                patient_birthDate = patient.birthDate
            except:
                pass
            # patinet_id, gender, birthDate
            patients_data.append([patient.id, patient.gender, patient_birthDate])
        patients_df = pd.DataFrame(patients_data, columns=["patient_id", "gender", "birthDate"])
        # Search for media
```

```
        media_list = client.resources('Media').include('Patient', 'subject')
        media_data = []
        for media in media_list:
            media_bodySite = None
            media_reasonCode = None
            media_note = None
            try:
                media_bodySite = media.bodySite.text
            except:
                pass
            try:
                media_reasonCode = media.reasonCode[0].text
            except:
                pass
            try:
                media_note = media.note[0].text
            except:
                pass
            media_data.append([media.subject.id, media.id, media_bodySite, media_reasonCode, media_note, media.content.url])
        media_df = pd.DataFrame(media_data, columns=["patient_id", "media_id", "bodySite", "reasonCode", "note", "image_url"])
        self.data_df = pd.merge(patients_df, media_df, on='patient_id', how='outer')
        self.data_df = self.data_df[self.data_df['note'].notna()].reset_index()
        self.trans = transforms.Compose([transforms.RandomHorizontalFlip(),
                                         transforms.RandomVerticalFlip(),
                                         transforms.ColorJitter(brightness=32. / 255., saturation=0.5),
                                         transforms.Resize(self.input_size),
                                         transforms.ToTensor()])

    def __len__(self):
        if self.split == "train":
            return int(len(self.data_df) * 0.8)
        elif self.split == "val":
            return int(len(self.data_df) * 0.2)

    def __getitem__(self, idx):
        val_start_id = int(len(self.data_df) * 0.8)
        if self.split == "train":
            idx = idx
        else:
            idx = idx + val_start_id
        img_url = self.data_df.loc[idx, 'image_url']
        note = self.data_df.loc[idx, 'note']
        y = self.classes[note]
        img_res = requests.get(img_url)
        if img_res.status_code == 200:
            x = Image.open(BytesIO(img_res.content))
            x = self.center_crop(x)
            x = self.trans(x)
            # Transform y
            y = np.int64(y)
            return {"image": x, "label": y}
        else:
            raise RuntimeError("Image url {} is not reachable!".format(img_url))
```

```python

    def center_crop(self, pil_img):
        img_width, img_height = pil_img.size
        if img_width > img_height:
            crop_size = img_height
        else:
            crop_size = img_width
        return pil_img.crop(((img_width - crop_size) // 2,
                             (img_height - crop_size) // 2,
                             (img_width + crop_size) // 2,
                             (img_height + crop_size) // 2))
```

Listing 1.3: isicdataset.py

```python
import os
import os.path as osp
import numpy as np

import glob
import tqdm
import shutil
import pytz
import datetime

import torch
import torch.nn as nn
from torch.autograd import Variable

from isicnet import ISICNet
from isicdataset import ISICDataset

## Define evaluation function
def _fast_hist(label_true, label_pred, n_class):
    hist = np.bincount(
        n_class * label_true.astype(int) +
        label_pred.astype(int), minlength=n_class ** 2).reshape(n_class, n_class)
    return hist

def label_accuracy_score(label_trues, label_preds, n_class=8):
    hist = np.zeros((n_class, n_class))
    hist += _fast_hist(label_trues, label_preds, n_class)
    acc = np.diag(hist).sum() / hist.sum()
    with np.errstate(divide='ignore', invalid='ignore'):
        precision = np.diag(hist) / hist.sum(axis=1)
    mean_precision = np.nanmean(precision)
    with np.errstate(divide='ignore', invalid='ignore'):
        recall = np.diag(hist) / hist.sum(axis=0)
    mean_recall = np.nanmean(recall)
    with np.errstate(divide='ignore', invalid='ignore'):
        iou = np.diag(hist) / (hist.sum(axis=1) + hist.sum(axis=0) - np.diag(hist))
    mean_iou = np.nanmean(iou)
    with np.errstate(divide='ignore', invalid='ignore'):
        f1 = (2 * np.diag(hist))/ (hist.sum(axis=1) + hist.sum(axis=0) + 2 * np.diag(hist))
    mean_f1 = np.nanmean(f1)
    return acc, mean_precision, mean_recall, mean_iou, mean_f1

## Define directory of output
here = osp.dirname(osp.abspath(__file__))
```

```
out_dir = osp.join(here, 'output')
if not os.path.exists(out_dir):
    os.makedirs(out_dir)
runs = sorted(glob.glob(os.path.join(out_dir, 'run_*')))
run_id = int(runs[-1].split('_')[-1]) + 1 if runs else 0
experiment_dir = os.path.join(out_dir, 'run_{}'.format(str(run_id)))
if not os.path.exists(experiment_dir):
    os.makedirs(experiment_dir)

## Define (input) variables from Docker Container environment variables
fhir_server = str(os.environ['FHIR_HOST_NAME'])
fhir_port = str(os.environ['FHIR_PORT'])
# num_station = int(os.environ['NUM_STATION'])
# sid = int(os.environ['SID'])
#Hyperparameters
batch_size = int(os.environ.get('BATCH_SIZE', '1'))
num_epoch = int(os.environ.get('NUM_EPOCH', '1'))
lr = float(os.environ.get('LR', '0.01'))
weight_decay = float(os.environ.get('WEIGHT_DECAY', '0.005'))
model_name = "resnet18"

## Define (output) file formats
if not osp.exists(osp.join(experiment_dir, 'val_log.csv')):
    with open(osp.join(experiment_dir, 'val_log.csv'), 'w') as f:
        header = ['epoch', 'Loss', 'Acc', 'Precision', 'Recall', 'Iou', '
    F1Score', 'train/Loss', 'elapsed_time']
        header = map(str, header)
        f.write(','.join(header) + '\n')
        print("Initial Log file")

cuda = torch.cuda.is_available()
torch.manual_seed(1337)
if cuda:
    torch.cuda.manual_seed(1337)

## Initial Model
print("Initial Model")
model = ISICNet(backbone=model_name)
print("Initial Model {}".format(model_name))
if cuda:
    print("Cuda:", cuda)
    model = model.cuda()
else:
    print("Running on CPU")
## Initial Datasets of train and val on station 1, 2, 3 and test
kwargs = {'num_workers': 4, 'pin_memory': True} if cuda else {}
print("Initial Training Dataset")
train_dataloader = torch.utils.data.DataLoader(ISICDataset(fhir_server,
    fhir_port, split='train'), batch_size=batch_size, shuffle=True, **
    kwargs)
print("Initial Val Dataset")
val_dataloader = torch.utils.data.DataLoader(ISICDataset(fhir_server,
    fhir_port, split='val'), batch_size=batch_size, shuffle=False, **kwargs
    )
## Initial criterion (Cross Entropy Loss)
print("Initial Loss function")
criterion = nn.CrossEntropyLoss()
## Initial Optimizers for station
print("Initial Optimizer")
optim = torch.optim.Adam(model.parameters(), lr=lr, weight_decay=
    weight_decay)
```

```python

## Load model from previous train
if run_id > 0:
    prev_experiment_dir = osp.join(out_dir, 'run_{}'.format(str(run_id - 1)))
    if osp.exists(osp.join(prev_experiment_dir, 'best_model.pth.tar')):
        prev_best_model = torch.load(osp.join(prev_experiment_dir, 'best_model.pth.tar'))
        model.load_state_dict(prev_best_model['model_state_dict'])
        optim.load_state_dict(prev_best_model['optim_state_dict'])
        shutil.copy(osp.join(prev_experiment_dir, 'best_model.pth.tar'),
                    osp.join(experiment_dir, 'best_model.pth.tar'))
        print("Model loaded from previous train.")
    else:
        print("No previous best model found!")
else:
    torch.save({
        'epoch': 0,
        'optim_state_dict': optim.state_dict(),
        'model_state_dict': model.state_dict(),
        'best_acc': 0.0,
    }, osp.join(experiment_dir, 'best_model.pth.tar'))

timestamp_start = datetime.datetime.now(pytz.timezone('Asia/Tokyo'))
best_acc = 0.0
## Run the training processing on the station
for epoch in range(num_epoch):
    model.train()
    train_loss = 0.0
    for batch_idx, sample in tqdm.tqdm(enumerate(train_dataloader), total=len(train_dataloader), desc='Station Train epoch=%d' % epoch, ncols=80, leave=False):
        assert model.training
        img, lbl = sample['image'], sample['label']
        if cuda:
            img, lbl = img.cuda(), lbl.cuda()
        img, lbl = Variable(img), Variable(lbl)
        optim.zero_grad()
        pred = model(img)
        loss = criterion(pred, lbl)
        train_loss = train_loss + loss.data.item()
        loss.backward()
        optim.step()

    train_loss = train_loss / len(train_dataloader)
    print("Train epoch {} finished with average train loss of {}.".format(epoch, train_loss))

    model.eval()
    val_loss = 0.0
    label_trues, label_preds = [], []
    for batch_idx, sample in tqdm.tqdm(enumerate(val_dataloader), total=len(val_dataloader), desc='Station Val epoch=%d' % epoch, ncols=80, leave=False):
        img, lbl = sample['image'], sample['label']
        if cuda:
            img, lbl = img.cuda(), lbl.cuda()
        img, lbl = Variable(img), Variable(lbl)
        with torch.no_grad():
            pred = model(img)
        loss = criterion(pred, lbl)
```

```
        val_loss = val_loss + loss.data.item()
        lbl = lbl.data.cpu().numpy()
        pred = pred.data.max(1)[1].cpu().numpy()
        label_trues = np.concatenate((label_trues, lbl), axis=0)
        label_preds = np.concatenate((label_preds, pred), axis=0)
    val_loss = val_loss / len(val_dataloader)
    acc, mean_precision, mean_recall, mean_iou, mean_f1 = label_accuracy_score(label_trues, label_preds)
    with open(osp.join(experiment_dir, 'val_log.csv'), 'a') as f:
        elapsed_time = (datetime.datetime.now(pytz.timezone('Asia/Tokyo')) - timestamp_start).total_seconds()
        log = [epoch, val_loss, acc, mean_precision, mean_recall, mean_iou, mean_f1, train_loss, elapsed_time]
        log = map(str, log)
        f.write(',').join(log) + '\n')

    is_best = acc > best_acc
    if is_best:
        best_acc = acc
    torch.save({
        'epoch': epoch,
        'optim_state_dict': optim.state_dict(),
        'model_state_dict': model.state_dict(),
        'best_acc': best_acc,
    }, osp.join(experiment_dir, 'checkpoint.pth.tar'))
    if is_best:
        shutil.copy(osp.join(experiment_dir, 'checkpoint.pth.tar'), osp.join(experiment_dir, 'best_model.pth.tar'))
    print("Station Val epoch {} finished with loss of {}, acc of {}, precision of {}, recall of {}, iou of {}, f1-score of {}.".format(epoch, val_loss, acc, mean_precision, mean_recall, mean_iou, mean_f1))
print("Finished training process")
```

Listing 1.4: main.py

```
import torch
import torch.nn as nn
from torchvision.models import resnet

class ISICNet(nn.Module):
    def __init__(self, n_feature=3, n_class=8, backbone='resnet18'):
        super(ISICNet, self).__init__()
        self.n_feature = n_feature
        self.n_class = n_class
        self.backbnone = backbone
        if self.backbnone == 'resnet18':
            base = resnet.resnet18(pretrained=True)
            self.resnet_expansion = 1
        self.in_block = nn.Sequential(
            nn.Conv2d(self.n_feature, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3), bias=False),
            base.bn1,
            base.relu,
            base.maxpool)
        self.encoder1 = base.layer1
        self.encoder2 = base.layer2
        self.encoder3 = base.layer3
        self.encoder4 = base.layer4
        self.avgpool = base.avgpool
        self.flatten = nn.Flatten()
        self.fc = nn.Linear(512*self.resnet_expansion, self.n_class , bias=
```

```
    True)

    def forward(self, x):
        h = self.in_block(x)
        h = self.encoder1(h)
        h = self.encoder2(h)
        h = self.encoder3(h)
        h = self.encoder4(h)
        y = self.fc(self.flatten(self.avgpool(h)))
        return y

if __name__ == "__main__":
    model = ISICNet().cuda()
    x = torch.rand(2, 3, 512, 512).cuda()
    y = model(x)
    print(y)
    print(y.data.max(1)[1])
    # t = torch.tensor([1.0, 0.0, 1.0])
    # y = torch.tensor([0.02, 0.05, 0.99])
    # bce = nn.BCELoss()
    # l = torch.log(torch.tensor(0.02))+ torch.log(torch.tensor(1-0.05)) +
    torch.log(torch.tensor(0.99))
    # print(bce(y, t))
    # print(l)
```

Listing 1.5: isicnet.py


```
{"resources":[{"id":"5bce186a-5005-4ca8-878e-9d8d4e0747c0","datasets":[{"id
    ":"0ed80d8f-6f34-45c7-979f-714681f59046a"},{"id":"a80bac0f-7f44-4fdd-
    a3c5-45f9aad175b6"}]}],"route":[{"id":"5bce186a-5005-4ca8-878e-9
    d8d4e0747c0","ownEnvs":[]}]}
```


Listing 1.6: route.json

1.5 Connection Parameters

Overall, 2 connection parameters have been found in the train:

Name	Type	Required?
FHIR_SERVER	string	True
FHIR_PORT	number	True

2. Static Train Analysis

Summary 2.1

- **Number of Lines:** 223754
- **Vulnerabilities per Line:** 0.0
- **Analysis Score:** 1.0

Total

Summary: Low: 0 Medium: 0 High: 0

2.1 SAST

Summary: Low: 0 Medium: 0 High: 0

Detailed Vulnerabilities:

Vulnerability 2.1

Test-specific ID: 62d9a0c9581c461f41278053d31b06a1ee7b663f285f97c18d2b90af109a170c
Severity: Info
Identifiers:

- **semgrep_id:** bandit.B101
- **cwe:** 754
- **bandit_test_id:** B101

Message:Improper Check for Unusual or Exceptional Conditions
File:ISIC2019II/main.py **Start line**: 128

2.2 Secret Detection

Summary: Low: 0 Medium: 0 High: 0

Detailed Secret Detection:

2.3 Dependency Analysis

Summary: Low: 0 Medium: 0 High: 0

Detailed Dependency Analysis:

2.4 Blacklist Detection

The following patterns have been configured:

`\bPOST\b`

The following matches have been found:

None

3. Train Image Analysis

3.1 Vulnerabilities

Summary: Low: 22 Medium: 8 High: 0 Critical: 0

Detailed Vulnerabilities:

Critical:

High:

Medium:

Vulnerability 3.1

Test-specific ID: SNYK-UBUNTU2004-GNUTLS28-6069042 **Severity:** medium
Identifiers: **Message**:Information Exposure

Vulnerability 3.2

Test-specific ID: SNYK-UBUNTU2004-GNUTLS28-6172192 **Severity:** medium
Identifiers: **Message**:Information Exposure

Vulnerability 3.3

Test-specific ID: SNYK-UBUNTU2004-NCURSES-5423141 **Severity:** medium
Identifiers: **Message**:Out-of-bounds Write

Vulnerability 3.4

Test-specific ID: SNYK-UBUNTU2004-OPENSSH-6142083 **Severity:** medium
Identifiers: **Message**:CVE-2023-51767

Vulnerability 3.5

Test-specific ID: SNYK-UBUNTU2004-OPENSSL-5661533 **Severity:** medium
Identifiers: **Message**:Allocation of Resources Without Limits or Throttling

Vulnerability 3.6

Test-specific ID: SNYK-UBUNTU2004-PAM-6170207 **Severity:** medium
Identifiers: **Message**:CVE-2024-22365

Vulnerability 3.7

Test-specific ID: SNYK-UBUNTU2004-TAR-6096088 **Severity:** medium
Identifiers: **Message**:CVE-2023-39804

Vulnerability 3.8

Test-specific ID: SNYK-UBUNTU2004-XZUTILS-5854646 **Severity:** medium
Identifiers: **Message**:CVE-2020-22916

Low:

Vulnerability 3.9

Test-specific ID: SNYK-UBUNTU2004-COREUTILS-583876 **Severity:** low
Identifiers: **Message**:Improper Input Validation

Vulnerability 3.10

Test-specific ID: SNYK-UBUNTU2004-GIT-580979 **Severity:** low
Identifiers: **Message**:Improper Input Validation

Vulnerability 3.11

Test-specific ID: SNYK-UBUNTU2004-GLIBC-1297554 **Severity:** low
Identifiers: **Message**:Use After Free

Vulnerability 3.12

Test-specific ID: SNYK-UBUNTU2004-GLIBC-2415100 **Severity:** low
Identifiers: **Message**:Allocation of Resources Without Limits or Throttling

Vulnerability 3.13

Test-specific ID: SNYK-UBUNTU2004-GLIBC-5901432 **Severity:** low
Identifiers: **Message**:Use After Free

Vulnerability 3.14

Test-specific ID: SNYK-UBUNTU2004-GLIBC-5901476 **Severity:** low
Identifiers: **Message**:Use After Free

Vulnerability 3.15

Test-specific ID: SNYK-UBUNTU2004-GNUPG2-3035407 **Severity:** low
Identifiers: **Message**:Out-of-bounds Write

Vulnerability 3.16

Test-specific ID: SNYK-UBUNTU2004-KRB5-579303 **Severity:** low
Identifiers: **Message**:Integer Overflow or Wraparound

Vulnerability 3.17

Test-specific ID: SNYK-UBUNTU2004-LIBPNG16-3124878 **Severity:** low
Identifiers: **Message**:NULL Pointer Dereference

Vulnerability 3.18

Test-specific ID: SNYK-UBUNTU2004-NCURSES-1656318 **Severity:** low
Identifiers: **Message**:Out-of-bounds Write

Vulnerability 3.19

Test-specific ID: SNYK-UBUNTU2004-NCURSES-2770341 **Severity:** low
Identifiers: **Message**:Out-of-bounds Read

Vulnerability 3.20

Test-specific ID: SNYK-UBUNTU2004-OPENSSH-1047872 **Severity:** low
Identifiers: **Message**:Information Exposure

Vulnerability 3.21

Test-specific ID: SNYK-UBUNTU2004-OPENSSL-5786273 **Severity:** low
Identifiers: **Message**:Inefficient Regular Expression Complexity

Vulnerability 3.22

Test-specific ID: SNYK-UBUNTU2004-OPENSSL-5811821 **Severity:** low
Identifiers: **Message**:Excessive Iteration

Vulnerability 3.23

Test-specific ID: SNYK-UBUNTU2004-PATCH-2325780 **Severity:** low
Identifiers: **Message**:Release of Invalid Pointer or Reference

Vulnerability 3.24

Test-specific ID: SNYK-UBUNTU2004-PATCH-582546 **Severity:** low
Identifiers: **Message**:Double Free

Vulnerability 3.25

Test-specific ID: SNYK-UBUNTU2004-PCRE3-580031 **Severity:** low
Identifiers: **Message**:Uncontrolled Recursion

Vulnerability 3.26

Test-specific ID: SNYK-UBUNTU2004-PROCPS-5816664 **Severity:** low
Identifiers: **Message**:Out-of-bounds Write

Vulnerability 3.27

Test-specific ID: SNYK-UBUNTU2004-SHADOW-5425687 **Severity:** low
Identifiers: **Message**:Arbitrary Code Injection

Vulnerability 3.28

Test-specific ID: SNYK-UBUNTU2004-SHADOW-577863 **Severity:** low
Identifiers: **Message**:Time-of-check Time-of-use (TOCTOU)

Vulnerability 3.29

Test-specific ID: SNYK-UBUNTU2004-SYSTEMD-3339226 **Severity:** low
Identifiers: **Message**:CVE-2023-26604

Vulnerability 3.30

Test-specific ID: SNYK-UBUNTU2004-SYSTEMD-6137854 **Severity:** low
Identifiers: **Message**:CVE-2023-7008

4. Dynamic Train Analysis

4.1 Image size differences after simulation:

4.1.1 Simulated route:

- Station 5bce186a-5005-4ca8-878e-9d8d4e0747c0: 225937765

4.1.2 Directory Overview of changed files:

Legend: Modified Added Deleted

```
Train Root
├── opt
│   └── conda
│       └── lib
│           └── python3.10
│               └── site-packages
│                   └── numpy
│                       └── __pycache__
│                           └── __config__.cpython-310.pyc
└── usr
    └── src
        └── app
            ├── __pycache__
            │   ├── isicdataset.cpython-310.pyc
            │   └── isicnet.cpython-310.pyc
            └── output
                └── run_0
                    ├── best_model.pth.tar
                    └── checkpoint.pth.tar
```

```
│   └── val_log.csv
└── root
    └── .cache
        └── torch
            └── hub
                └── checkpoints
                    └── resnet18-f37072fd.pth
```

4.1.3 ./simulationResults/__config__.cpython-310.pyc

No Preview Available! `https://git.rwth-aachen.de/padme-development/padme-train-depot/-/jobs/4570446/artifacts/file/simulationResults/__config__.cpython-310.pyc`

4.1.4 ./simulationResults/isicdataset.cpython-310.pyc

No Preview Available! `https://git.rwth-aachen.de/padme-development/padme-train-depot/-/jobs/4570446/artifacts/file/simulationResults/isicdataset.cpython-310.pyc`

4.1.5 ./simulationResults/isicnet.cpython-310.pyc

No Preview Available! `https://git.rwth-aachen.de/padme-development/padme-train-depot/-/jobs/4570446/artifacts/file/simulationResults/isicnet.cpython-310.pyc`

4.1.6 ./simulationResults/best_model.pth.tar

No Preview Available! `https://git.rwth-aachen.de/padme-development/padme-train-depot/-/jobs/4570446/artifacts/file/simulationResults/best_model.pth.tar`

4.1.7 ./simulationResults/checkpoint.pth.tar

No Preview Available! `https://git.rwth-aachen.de/padme-development/padme-train-depot/-/jobs/4570446/artifacts/file/simulationResults/checkpoint.pth.tar`

4.1.8 ./simulationResults/val_log.csv

Listing 4.1: val_log.csv

```
epoch,Loss,Acc,Precision,Recall,Iou,F1Score,train/Loss,
    elapsed_time
0,714102.0,0.0,0.0,0.0,0.0,0.0,7.464000657200813,1.320086
```

4.1.9 ./simulationResults/resnet18-f37072fd.pth

No Preview Available! `https://git.rwth-aachen.de/padme-development/padme-train-depot/-/jobs/4570446/artifacts/file/simulationResults/resnet18-f37072fd.pth`

4.2 Execution metrics:

- Maximum CPU Usage:5.8371202223355425
- Maximum Memory Usage:562102272
- Maximum Number of PIDS:11

4.3 Network I/O:

- RX (Reading):47501630
- TX (Transmitting):340878

5. Conclusion

The PADME Security Audit has come to the following conclusion:

Static Complete	SAST	Dependencies	Secret Detection

Standard Compliant	Blacklist	Train Image Analysis	DAST

Warnings

- Identified network traffic in dynamic analysis.
- Couldn't detect compliance with standards (usage of the python module padme_conductor [https://pypi.org/project/padme-conductor]).

Audit Result

Train Rejected

January 31, 2024