



Security Insider Lab II

## Second Lab - Creating LSTM Model

Sayed Alisina Qaderi, Atiqullah Ahmadzai & Dusan Dordevic

---

15. Mai 2024

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Methods</b>	<b>2</b>
2.0.1	Data Collection . . . . .	2
2.0.2	Keyword Filtering . . . . .	3
2.0.3	Language Segregation . . . . .	3
2.0.4	Commit Analysis . . . . .	4
2.0.5	Dataset Acquisition . . . . .	5
2.0.6	Model Training . . . . .	5
<b>3</b>	<b>Results</b>	<b>9</b>
3.0.1	Data Collection . . . . .	9
3.0.2	Keyword Filtering . . . . .	9
3.0.3	Language Segregation . . . . .	9
3.0.4	Commit Analysis . . . . .	12
3.0.5	Model Training . . . . .	12
<b>4</b>	<b>Discussion</b>	<b>14</b>
4.0.1	Creating the LSTM models . . . . .	14

# List of Figures

2.1	Gathering repositories . . . . .	3
2.2	GetDiffs missing os . . . . .	4
2.3	GetDiffs typo issue . . . . .	4
2.4	Sklearn Utils Class Weight . . . . .	6
2.5	Yhat Changes . . . . .	6
2.6	Loading Data, Filtering Array & Perpairing Data . . . . .	7
2.7	The Epoch Steps Calculation . . . . .	7
2.8	Model Generation and Saving it . . . . .	8
3.1	Scrapped Repositories . . . . .	10
3.2	Showcases filtered repositories . . . . .	10
3.3	No showcases filtered repositories . . . . .	11
3.4	Repositories with Python code . . . . .	11
3.5	Repositories without Python code . . . . .	12
3.6	PyCommitsWithDiffs data . . . . .	12

# 1 Introduction

This report documents the third exercise of Security Insider Lab 2, detailing our process in training our model and creating the dataset. The exercise comprised six key steps.

**Data Collection :** We initiated the process by scraping GitHub and gathering data from 2000 repositories.

**Keyword Filtering:** Next, we examined the names and readme files of the scraped repositories, filtering out those that aligned with our predefined keywords.

**Language Segregation:** Our third step involved segregating libraries with Python code from those without.

**Commit Analysis:** Subsequently, we meticulously traversed the filtered repositories, extracting all differences from their commits.

**Dataset Acquisition:** In the fifth step, we procured the necessary dataset provided by the central repository, which is crucial for subsequent stages.

**Model Training:** Finally, we proceeded to train our model based on the acquired dataset from the previous step, completing the exercise.

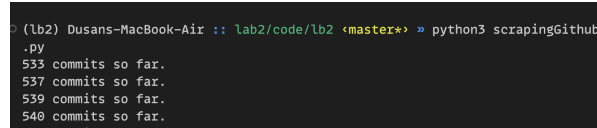
## 2 Methods

### 2.0.1 Data Collection

We initiated the data collection process by scraping GitHub repositories. Utilizing web scraping techniques, we gathered data from 2000 repositories, encompassing a diverse range of projects. The default amount of repositories was 100000, which is a considerable amount. Considering our time constraints and the need for a more efficient process, we made the decision to change the default steps to 2000, keeping you informed and involved in the process. 2.1

As we worked on **scrapingGithub.py** file, we have faced the following issues. The solutions are also mentioned in below:

- Missing **requests** library solved with *pip install requests*
- Missing **requests\_oauthlib** library, solved with *pip install requests\_oauthlib*
- Missing **all\_commits** file which we created manually.
- Missing Github Access Token, we generated through Github and save to **access** file inside the project folder.



```
(lb2) Dusan-MacBook-Air :: lab2/code/lb2 <master> » python3 scrapingGithub.py
533 commits so far.
537 commits so far.
539 commits so far.
540 commits so far.
```

Figure 2.1: Gathering repositories

- Requests needed exception handling in case of losing the network.

After solving the above issue, they successfully ran the **scrapingGithub.py**, and its results were saved in **allcommits.json** file.

### 2.0.2 Keyword Filtering

Following data collection, we performed keyword filtering to refine the dataset with **filterShowcase.py** file. By examining the names and readme files of the scraped repositories, we identified and filtered out repositories that matched our predefined keywords. We have filtered the following showcases:

- `toomuchsecurity = ['offensive', 'pentest', 'vulnerab', 'security', 'hack', 'exploit', 'ctf ', ' ctf', 'capture the flag','attack']`
- `alittletoomuch = ['offensive security', 'pentest', 'exploits', 'vulnerability research', 'hacking', 'security framework', 'vulnerability database', 'simulated attack', 'security research']`

### 2.0.3 Language Segregation

With the help of **getDiffs.py** class file, we refined the dataset by segregating libraries with Python code from those without. This step was crucial for our subsequent analysis,

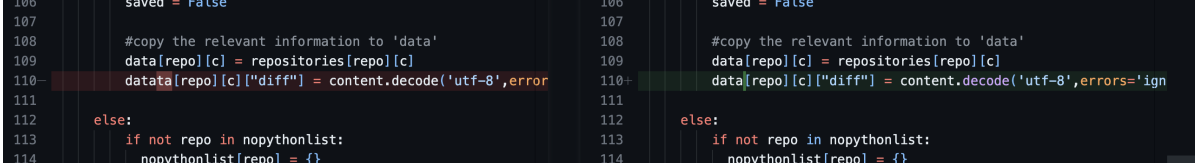
## 2 Methods



```
Code > getDiffs.py > ...
1 import requests
2 import time
3 import sys
4 import json
5 import datetime
6
7

1 import requests
2 import time
3 import sys
4 import json
5 import datetime
6 import os
7
```

Figure 2.2: GetDiffs missing os



```
106 saved = False
107
108 #copy the relevant information to 'data'
109 data[repo][c] = repositories[repo][c]
110 datata[repo][c]["diff"] = content.decode('utf-8',error
111
112 else:
113     if not repo in nopythonlist:
114         nopythonlist[repo] = {}

106 saved = False
107
108 #copy the relevant information to 'data'
109 data[repo][c] = repositories[repo][c]
110 data[repo][c]["diff"] = content.decode('utf-8',errors='ign
111
112 else:
113     if not repo in nopythonlist:
114         nopythonlist[repo] = {}
```

Figure 2.3: GetDiffs typo issue

focusing specifically on repositories containing Python code. During this process, we faced the following issues:

- missing os library which resolved by *import os* 2.2.
- **datata** typo in line 110, which resolved by changing to **data** 2.3.

### 2.0.4 Commit Analysis

In this step, we meticulously analyzed and downloaded the commits of the filtered repositories with the help of the **getData.py** file. By extracting all differences from their commits, we gained insights into the projects' evolution and the changes made over time. We faced with only one issue in this phase *ModuleNotFoundError: No module named 'keras.layers.convolutional'* which resolved by *pip install keras* and the rest of code worked ideally and saved the result in **PyCommitsWithDiffs.json** JSON file.

### 2.0.5 Dataset Acquisition

To augment our dataset, we procured additional data from the main repository. This supplementary dataset, provided by the main repository, was essential for enhancing the depth and scope of our analysis in subsequent stages.

### 2.0.6 Model Training

The final step involved training our model based on the augmented dataset acquired in the previous step. This step handled with the help **makemodel.py**. The Python script is a machine learning pipeline for training and evaluating an LSTM (Long Short-Term Memory) model for identifying vulnerabilities. We have trained the model over four types of vulnerabilities, XSS (Cross-Site Scripting), Path Disclosure, Remote Code Execution. The model specification is 10 for minimum count, 200 for iterations, 300 for vector size.

While working with this script we have faced the following issues due to library deprecations and changes in the Python version:

1. **Sklearn.util class\_\_weight** issue which the input parameters changed in the latest versions. The differences shown in the figure 2.4.
2. **AttributeError: The vocab attribute was removed from KeyedVector in Gensim 4.0.0.**, the solution is changing vocab to **key\_to\_index**.
3. **vector = w2v\_model[t] 'Word2Vec' object is not subscriptable** resolved by changing *w2v\_model* to *m2v\_model.wv*.



```
# Before the changes
# class_weights = class_weight.compute_class_weight('balanced', numpy.unique(y_train), y_train)

# After the changes
classes = numpy.unique(y_train)
class_weights = class_weight.compute_class_weight(class_weight='balanced', classes=classes, y=y_train)
```

Figure 2.4: Sklearn Utils Class Weight

```
predict=model.predict(X_train)
yhat_classes=numpy.argmax(predict,axis=1)

#yhat_classes = model.predict_classes(X_train, verbose=0)
accuracy = accuracy_score(y_train, yhat_classes)
```

Figure 2.5: Yhat Changes

4. `yhat_classes = model.predict_classes(X_train, verbose=0)` issue because of the ... add in here . Therefore we have changed to the `predict=model.predict(X_train)` `yhat_classes=numpy.argmax(predict,axis=1)`. The final result for this changes shown in figure 2.5.
5. `numpy.core._exceptions._arraymemoryerror: unable to allocate 2.42 gib for an array with shape (8277, 131, 300) and data type float64` memory issue, we solved this issue with increasing virtual memory to 150 GB and GPU with the help of Tensorflow GPU library.

Figures 2.6, 2.7 and 2.8 log of each step for execution of this step.

MARK TODO: Talk About the code enhancemnt you have and check if I miss something with ATIQullah

## 2 Methods

```
[env] (base) ~/lilina@SomeOne: Code % python makemodel.py
output/models/word2vec_withString10-200-300.model
finished loading. 11:22
cutoff 38625
cutoff2 46982
Creating training dataset... (xss)
Creating validation dataset...
Creating finaltest dataset...
Train length: 38625
Test length: 8277
Finaltesting Length: 8277
Time: 11:27
38625
numpy array done. 11:30
38625 samples in the training set.
8277 samples in the validation set.
8277 samples in the final test set.
percentage of vulnerable samples: 8.78%
absolute amount of vulnerable samples in test set: 719
Starting LSTM: 11:30
Dropout: 0.2
Neurons: 180
Optimizer: adam
Epochs: 100
Batch Size: 128
Max length: 200
2024-05-15 11:31:21.811590: I tensorflow/core/device/metal_device.cc:1154] Metal device set to: Apple M1 Pro
2024-05-15 11:31:21.812643: I tensorflow/core/device/metal_device.cc:296] SystemMemory: 16.00 GB
2024-05-15 11:31:21.812901: I tensorflow/core/device/metal_device.cc:313] MaxCacheSize: 5.33 GB
2024-05-15 11:31:21.813483: I tensorflow/core/common_runtime/pluggable_device/pluggable_device_factory.cc:385] Could not identify NPU node of platform GPU ID 0, defaulting to 0. Your kernel may not have been built with NPU support.
2024-05-15 11:31:21.813718: I tensorflow/core/common_runtime/pluggable_device/pluggable_device_factory.cc:271] Created TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 with 0 MB memory) -> physical PluggableDevice (devi
ce: 0, name: METAL, pci bus id: <undefined>)
/Users/~/lilina@someone~/lib/python3.11/site-packages/keras/src/layers/core/dense.py:87: UserWarning: Do not pass an 'input_shape' / 'input_dim' argument to a layer. When using Sequential models, prefer using an 'Input(shape)' object as
the first layer in the model instead.
super().__init__(activity_regularizer=activity_regularizer, **kwargs)
Compiled LSTM: 11:31
Epoch 1/100
2024-05-15 11:31:26.197976: I tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:117] Plugin optimizer for device_type GPU is enabled.
302/382 ----- 7s 12ms/step - f1: 12.0216 - loss: 0.7382
Epoch 2/100
302/382 ----- 3s 10ms/step - f1: 8.5568 - loss: 0.6843
Epoch 3/100
302/382 ----- 3s 10ms/step - f1: 7.1469 - loss: 0.5573
Epoch 4/100
302/382 ----- 3s 10ms/step - f1: 7.5964 - loss: 0.5437
Epoch 5/100
302/382 ----- 3s 10ms/step - f1: 7.2558 - loss: 0.5352
Epoch 6/100
302/382 ----- 3s 10ms/step - f1: 7.2409 - loss: 0.5375
Epoch 7/100
302/382 ----- 3s 10ms/step - f1: 7.0938 - loss: 0.5874
Epoch 8/100
302/382 ----- 3s 10ms/step - f1: 7.0236 - loss: 0.5836
Epoch 9/100
302/382 ----- 3s 10ms/step - f1: 7.4462 - loss: 0.5118
Epoch 10/100
302/382 ----- 3s 10ms/step - f1: 7.0716 - loss: 0.5818
Epoch 11/100
302/382 ----- 3s 10ms/step - f1: 7.0598 - loss: 0.4910
Epoch 12/100
302/382 ----- 3s 10ms/step - f1: 7.0600 - loss: 0.4830
Epoch 13/100
302/382 ----- 3s 10ms/step - f1: 7.1833 - loss: 0.4793
Epoch 14/100
302/382 ----- 3s 11ms/step - f1: 7.4353 - loss: 0.4165
```

Figure 2.6: Loading Data, Filtering Array & Perpairing Data

```
302/382 ----- 3s 11ms/step - f1: 7.4353 - loss: 0.4765
Epoch 15/100
302/382 ----- 3s 11ms/step - f1: 7.3694 - loss: 0.4760
Epoch 16/100
302/382 ----- 3s 10ms/step - f1: 7.3847 - loss: 0.4697
Epoch 17/100
302/382 ----- 3s 10ms/step - f1: 7.4753 - loss: 0.4767
Epoch 18/100
302/382 ----- 3s 10ms/step - f1: 7.2681 - loss: 0.4733
Epoch 19/100
302/382 ----- 3s 10ms/step - f1: 7.4522 - loss: 0.4861
Epoch 20/100
302/382 ----- 3s 11ms/step - f1: 7.4318 - loss: 0.4629
Epoch 21/100
302/382 ----- 3s 11ms/step - f1: 7.5971 - loss: 0.4634
Epoch 22/100
302/382 ----- 3s 11ms/step - f1: 7.5129 - loss: 0.4685
Epoch 23/100
302/382 ----- 3s 11ms/step - f1: 7.6544 - loss: 0.4833
Epoch 24/100
302/382 ----- 3s 11ms/step - f1: 7.2577 - loss: 0.4481
Epoch 25/100
302/382 ----- 3s 11ms/step - f1: 6.9480 - loss: 0.4622
Epoch 26/100
302/382 ----- 3s 11ms/step - f1: 7.4185 - loss: 0.4607
Epoch 27/100
302/382 ----- 3s 11ms/step - f1: 6.8265 - loss: 0.4574
Epoch 28/100
302/382 ----- 3s 11ms/step - f1: 7.1641 - loss: 0.4531
Epoch 29/100
302/382 ----- 3s 11ms/step - f1: 7.3951 - loss: 0.4537
Epoch 30/100
302/382 ----- 4s 12ms/step - f1: 7.9858 - loss: 0.4665
Epoch 31/100
302/382 ----- 4s 11ms/step - f1: 7.5451 - loss: 0.4801
Epoch 32/100
302/382 ----- 3s 11ms/step - f1: 7.6279 - loss: 0.4532
Epoch 33/100
302/382 ----- 3s 11ms/step - f1: 7.4836 - loss: 0.4630
Epoch 34/100
302/382 ----- 3s 11ms/step - f1: 7.4816 - loss: 0.4516
Epoch 35/100
302/382 ----- 3s 11ms/step - f1: 7.3474 - loss: 0.4481
Epoch 36/100
302/382 ----- 3s 11ms/step - f1: 7.4998 - loss: 0.4559
Epoch 37/100
302/382 ----- 3s 11ms/step - f1: 7.3881 - loss: 0.4625
Epoch 38/100
302/382 ----- 3s 11ms/step - f1: 7.3695 - loss: 0.4564
Epoch 39/100
302/382 ----- 3s 11ms/step - f1: 7.2762 - loss: 0.4414
Epoch 40/100
302/382 ----- 3s 11ms/step - f1: 7.3555 - loss: 0.4683
Epoch 41/100
302/382 ----- 3s 11ms/step - f1: 7.3459 - loss: 0.4586
Epoch 42/100
302/382 ----- 3s 11ms/step - f1: 7.5433 - loss: 0.4488
Epoch 43/100
302/382 ----- 3s 11ms/step - f1: 7.0451 - loss: 0.4457
Epoch 44/100
302/382 ----- 3s 11ms/step - f1: 7.1964 - loss: 0.4558
Epoch 45/100
302/382 ----- 3s 11ms/step - f1: 7.0636 - loss: 0.4373
Epoch 46/100
302/382 ----- 3s 11ms/step - f1: 6.8576 - loss: 0.4384
```

Figure 2.7: The Epoch Steps Calculation

## 2 Methods

```
Epoch 84/100      4s 12ms/step - f1: 7.4266 - loss: 0.4147
302/302
Epoch 85/100      3s 11ms/step - f1: 7.3414 - loss: 0.4404
302/302
Epoch 86/100      3s 11ms/step - f1: 7.3580 - loss: 0.4381
302/302
Epoch 87/100      3s 11ms/step - f1: 7.1336 - loss: 0.4345
302/302
Epoch 88/100      3s 11ms/step - f1: 7.0772 - loss: 0.4572
302/302
Epoch 89/100      4s 12ms/step - f1: 7.0628 - loss: 0.4441
302/302
Epoch 90/100      4s 12ms/step - f1: 7.1329 - loss: 0.4474
302/302
Epoch 91/100      4s 12ms/step - f1: 7.2380 - loss: 0.4400
302/302
Epoch 92/100      4s 12ms/step - f1: 7.1454 - loss: 0.4577
302/302
Epoch 93/100      4s 12ms/step - f1: 7.6336 - loss: 0.4445
302/302
Epoch 94/100      4s 12ms/step - f1: 7.4493 - loss: 0.4692
302/302
Epoch 95/100      4s 12ms/step - f1: 7.4182 - loss: 0.4265
302/302
Epoch 96/100      4s 13ms/step - f1: 7.2182 - loss: 0.4419
302/302
Epoch 97/100      4s 12ms/step - f1: 7.0694 - loss: 0.4335
302/302
Epoch 98/100      4s 12ms/step - f1: 7.1132 - loss: 0.4450
302/302
Epoch 99/100      4s 13ms/step - f1: 7.1951 - loss: 0.4445
302/302
Epoch 100/100     4s 12ms/step - f1: 7.3514 - loss: 0.4318
302/302
Now predicting on train set (0.2 dropout)
1708/1200      4s 3ms/step
Accuracy: 0.9121294498381877
Precision: 0.9568647249198939
Recall: 0.5
F1 score: 0.477023

Now predicting on test set (0.2 dropout)
259/259      1s 2ms/step
Accuracy: 0.9131327775764165
Precision: 0.9565663887882883
Recall: 0.5
F1 score: 0.477297

Now predicting on finaltest set (0.2 dropout)
259/259      1s 2ms/step
Accuracy: 0.9113205267689837
Precision: 0.9556602633884518
Recall: 0.5
F1 score: 0.476882

saving LSTM model xss: 11:37
WARNING:absl:You are saving your model as an HDF5 file via 'model.save()' or 'keras.saving.save_model(model)'. This file format is considered legacy. We recommend using instead the native Keras format, e.g. 'model.save('my_model.keras')' or 'keras.saving.save_model(model, 'my_model.keras')'.
```

Figure 2.8: Model Generation and Saving it

## 3 Results

### 3.0.1 Data Collection

All results are saved in `allcommits.json()` 3.1 file that we are required to create prior to executing the script. If the file is not created accordingly, the following error message is printed in the terminal: "The file is empty or does not exist."

### 3.0.2 Keyword Filtering

The result of `filterShowcase.py` is saved to the **DataFilter.json** file. It contains two JSON arrays for handling showcases and not showcases. The result is shown in figures 3.2 and 3.2.

### 3.0.3 Language Segregation

The result, which separates those repositories with Python code from those without Python code, is also saved in the **DataFilter.json** file. The results have been saved in two separated JSON arrays with names **no-python** and **python**, as shown in Figures 3.4 and 3.5.

### 3 Results

```
Code > {} all_commits.json > ...
1 {"https://github.com/openucx/ucx": {"98b8e8c0c9722541485f7a4efde59dbc3b29eba8": {"url": "https://api.github.com/repos/openucx/ucx/commits/98b8e8c0c9722541485f7a4efde59dbc3b29eba8", "sha": "98b8e8c0c9722541485f7a4efde59dbc3b29eba8", "keyword": "buffer overflow prevent"}, "d66092921c073838ee1670e2530151acfea2ebde": {"url": "https://api.github.com/repos/openucx/ucx/commits/d66092921c073838ee1670e2530151acfea2ebde", "html_url": "https://github.com/openucx/ucx/commit/d66092921c073838ee1670e2530151acfea2ebde", "sha": "d66092921c073838ee1670e2530151acfea2ebde", "keyword": "buffer overflow prevent"}}, "https://github.com/irontec/sngrep": {"f3f8ed8ef38748e6d61044b39b0dabd7e37c6809": {"url": "https://api.github.com/repos/irontec/sngrep/commits/f3f8ed8ef38748e6d61044b39b0dabd7e37c6809", "html_url": "https://github.com/irontec/sngrep/commit/f3f8ed8ef38748e6d61044b39b0dabd7e37c6809", "message": "Fix Buffer Overflow in SIP Header Processing\n\nResolved a critical buffer overflow in handling \"Call-ID\" and \"X-Call-ID\" SIP headers. This patch adds bounds checking and ensures string null-termination, preventing potential arbitrary code execution or DoS from malformed SIP messages.", "sha": "f3f8ed8ef38748e6d61044b39b0dabd7e37c6809", "keyword": "buffer overflow prevent"}}, "https://github.com/ericgoins/vifm": {"d0d27d206d9794e82ce578e59b4d3353c569699e": {"url": "https://api.github.com/repos/ericgoins/vifm/commits/d0d27d206d9794e82ce578e59b4d3353c569699e", "html_url": "https://github.com/ericgoins/vifm/commit/d0d27d206d9794e82ce578e59b4d3353c569699e", "message": "Fix a couple of buffer overflow warnings\n\nThey prevent optimized build with -Werror.", "sha": "d0d27d206d9794e82ce578e59b4d3353c569699e", "keyword": "buffer overflow prevent"}, "5cfd9cf0fa6599b1051cdd5b87166ea036749654": {"url": "https://api.github.com/repos/ericgoins/vifm/commits/5cfd9cf0fa6599b1051cdd5b87166ea036749654", "html_url": "https://github.com/ericgoins/vifm/commit/5cfd9cf0fa6599b1051cdd5b87166ea036749654", "sha": "5cfd9cf0fa6599b1051cdd5b87166ea036749654", "keyword": "buffer overflow fix"}, "e55e42aeb8393b3085caa49a86f30f68a1a3e952": {"url": "https://api.github.com/repos/ericgoins/vifm/commits/e55e42aeb8393b3085caa49a86f30f68a1a3e952", "html_url": "https://github.com/ericgoins/vifm/commit/e55e42aeb8393b3085caa49a86f30f68a1a3e952", "sha": "e55e42aeb8393b3085caa49a86f30f68a1a3e952", "keyword": "buffer overflow prevent"}}
```

Figure 3.1: Scrapped Repositories

```
{
  "showcase": {
    "nethackathon/nethackathon-nethack": {},
    "elfmz/far2l": {},
    "TheBombSquad/SMB2WorkshopMod": {},
    "ccavxx/exploit-db": {},
    "arancour69/exploit-database": {},
    "artyang/exploitdb": {},
    "merlinepedra25/EXPLOITDB": {},
    "razortag97/offsec_exploits": {},
    "conan25216/exploitdb": {},
    "chris-0x01/https-gitlab.com-exploit-database-exploitdb": {},
    "Jaboox/offensive-security-exploit-database": {},
    "michael101096/offensive-security-exploitdb": {},
    "jameser/exploitdb-reduced": {},
    "Brocks-Collections/exploit-database": {},
    "NoorahSmith/Exploit-DB-offsec": {},
    "sawhtetkhine-soe/exploit-database": {},
    "HackYourShit/exploit-database": {},
    "ubboolean/exploitdb": {},
    "chushuai/webappsecurity": {},
    "anhilo/exploit-database": {},
    "3v1lW1th1n/exploit-exploitdb": {},
    "offensive-security/exploitdb": {},
    "EthicalSecurity-Agency/exploit-database-exploitdb": {},
    "xianlimei/exploitdb": {},
    "jhe8281/exploitdb": {},
    "merlinepedra25/EXPLOITDB": {},
    "SYNgularity1/exploits": {},
    "bsd-hacker/freebsd": {},
    "brain-hackers/buildroot": {},
    "slsa-framework/slsa-verifier": {},
    "hackagadget/tarfs": {},
    "Innovation-Web-3-0-Blockchain/Hacking-Smart-Contracts": {},
    "p0-security/iam-privilege-catalog": {},
    "jitsecurity-soss/langchain": {},
    "jitsecurity-soss/python-code-": {},
    "H4lo/awesome-IoT-security-article": {},
    "mullvad/mullvadvpn-app": {},
    "derekarends/solidity-vulnerabilities": {}
  }
}
```

Figure 3.2: Showcases filtered repositories

### 3 Results

```
"showcase": {--
},
"noshowcase": {
  "openucx/ucx": {},
  "ironotec/sngrep": {},
  "ericgoins/vifm": {},
  "Arena-Rosnav/rosnav-rl": {},
  "2ndBaseChris/vifmChallenge": {},
  "thesourcerer8/hddsupliclone": {},
  "selimvuz/Library-Automation-System-in-C": {},
  "golded-plus/golded-plus": {},
  "Ridderrasmus/RPVoiceChat": {},
  "blooo-io/LedgerHQ-app-plugin-THORSwap": {},
  "nrfconnect/sdk-nrf": {},
  "nikso-itu/duckdb-oml": {},
  "netdata/netdata": {},
  "qwx9/syro": {},
  "RealYukiSan/chat_app": {},
  "dtrebilco/PortalsSokol": {},
  "Chysn/VIC20-wAx2": {},
  "memoryhole/libkiss": {},
  "rcjcooke/apfc": {},
  "google-deepmind/mujoco": {},
  "saprykin/plibsys": {},
  "quectel-official/QModemHelper": {},
  "gonoso/blender": {},
  "Bforartists/Bforartists": {},
  "UPBGE/upbge": {},
  "mrohne/ngspice": {},
  "ridobe/ridobe-seiei": {},
  "imr/ngspice": {},
  "amalej/onfire-cli": {},
  "RobertONelson/linux-stable-rcn-ee": {},
  "TaranovK/linuxnext": {},
  "Aquatic-Symbiosis-Genomics-Project/curation_tool": {},
  "torvalds/linux": {},
  "alexzahnaudio/PFM10": {},
  "casept/linux-samsung-smartwatch": {},
  "xu1119/torvalds-linux": {},
```

Figure 3.3: No showcases filtered repositories

```
Code > {} DataFilter.json > {} showcase
1 |
2 > "showcase": {--
44 | },
45 > "noshowcase": {--
1487 | },
1488 > "no-python": {--
2795 | },
2796 | "python": {
2797 |   "Arena-Rosnav/rosnav-rl": {},
2798 |   "recursionpharma/gflownet": {},
2799 |   "theroyallab/tabbyAPI": {},
2800 |   "amiyuki7/ethics_game": {},
2801 |   "mr-mdd/C5CK541---EOM---Group-A": {},
2802 |   "2lambda123/https-github.com-NationalSecurityAgency-ghidra": {},
2803 |   "salukadev/guide_app_rpi": {},
2804 |   "sab-rinakl/ctf": {},
2805 |   "Freedom-of-Form-Foundation/FFFTail": {},
2806 |   "securesauce/precli": {},
2807 |   "NetBSD/pkgsrc": {},
2808 |   "DL124/aaf-external": {},
```

Figure 3.4: Repositories with Python code

### 3 Results

```
Code > {} DataFilter.json > {} showcase

1  {
2  >   "showcase": { -
44  },
45  >   "noshowcase": { -
1487  },
1488   "no-python": {
1489     "openucx/ucx": {},
1490     "irontec/sngrep": {},
1491     "ericgoins/vifm": {},
1492     "2ndBaseChris/vifmChallenge": {},
1493     "thesourcerer8/hddsupliclone": {},
1494     "selimvuz/Library-Automation-System-in-C": {},
1495     "golded-plus/golded-plus": {},
1496     "Ridderrasmus/RPVoiceChat": {},
1497     "blooo-io/LedgerHQ-app-plugin-THORSwap": {},
1498     "nrfconnect/sdk-nrf": {},
1499     "nikso-itu/duckdb-oml": {},
1500     "netdata/netdata": {},
1501     "qwx9/syro": {},
1502     "RealYukiSan/chat_app": {}.
```

Figure 3.5: Repositories without Python code

[illegible]

Figure 3.6: PyCommitsWithDiffs data

### 3.0.4 Commit Analysis

As mentioned before, the result of this step has been saved in the **PyCommitsWithDiffs.json** JSON file. The files are structured as JSON objects; each object is a repository, and commit diffs are located inside that. Figure 3.6 shows a sample of this step result.

### 3.0.5 Model Training

We have trained the model > add detail in here

The final results are: and here

Table 3.1: Performance Metrics

Model	Accuracy	Precision	Recall	F1 Score
<b>XSS</b>				
Train Set	0.912	0.956	0.500	0.477
Test Set	0.913	0.957	0.500	0.477
Final Test Set	0.911	0.956	0.500	0.477
<b>Path Disclosure</b>				
Train Set	0.884	0.942	0.500	0.469
Test Set	0.883	0.941	0.500	0.469
Final Test Set	0.883	0.941	0.500	0.469
<b>Remote Code Execution</b>				
Train Set	0.910	0.955	0.500	0.477
Test Set	0.913	0.956	0.500	0.477
Final Test Set	0.909	0.955	0.500	0.476
<b>Command Injection</b>				
Train Set	0.896	0.936	0.478	0.491
Test Set	0.901	0.943	0.481	0.498
Final Test Set	0.893	0.935	0.479	0.493



## 4 Discussion

### 4.0.1 Creating the LSTM models

**Makemodel.py** script is used for creating data models. It splits the data in 3 different segments (train, validate, final). On the line of code 134, the samples are randomized with **for** loop. However, there are certain issues regarding this code. Library **Keras** that is imported in **myutils.py** file is causing a build error. Keras package is now **included** in previous installation of tensorflow package. Therefore, imports needed to be adjusted accordingly. "from keras.datasets import imdb" is now "tensorflow.keras.datasets". Also, we need to install all necessary packages (gensim, sklearn, tensorflow). Next, datasets are created using word2vec models created in previous exercise. Unfortunately, new error occurs where we need to adjust `if t in word_vectors.key_to_index and t != " ":` this line of code to match the new key to index parameter. Console is clear from errors so we can proceed to make our new vulnerability model. After loading the data, script creates new training dataset `print("Creating training dataset... (" + mode + ")")`, followed by `print("Creating validation dataset...")` and finally `print("Creating finaltest dataset...")`.