**Experiment & Results**

**Experiment 1 - Features**

The dataset we used in our experiment is the famous Drebin Malware Dataset. To use this dataset we need to manually select features then train our model on it. To find the optimum features in our use case we used a feature selection algorithm to justify the priority of features to detect a malware.

**Experiment 1.1 - Feature Selection**

We build a classification task using 3 informative features by building a forest and compute the feature importances, then select the top three ranked feature importances of the forest. We used a Random Forest Classifier model for this task. With no class weights and selected criterion function as ‘Gini’ and an n estimator of 1000. The result is shown below.

| Feature | Rank (score) |
| --- | --- |
| Opcode | 1 (0.271525 |
| Real Permissions | 2 (0.146973) |
| API Call | 3 (0.125988) |
| Intent | 4 (0.108765) |
| Service | 5 (0.107854) |
| Service\_Receiver | 6 (0.107013) |
| Provider | 7 (0.069560) |
| URL | 8 (0.062323) |

**Experiment 1.2 - Feature Extraction**

From each app in our datasets the following raw input feature sets are extracted (which are previously identified as effective) :

(1) opcode instruction sequences

(2) Android permissions usage

(3) Arbitrary and Proprietary Android API package

**Experiment 2 - Random Forest**

Random Forest Classifier is a classification algorithm that is made up of several decision trees. Each individual tree is built using randomness to promote uncorrelated forests, which then uses the predictive powers of the forest to make accurate decisions. It is a meta estimator that fits multiple decision tree classifiers on a number of sub-samples of the dataset. Also, to improve the predictive accuracy and control over-fitting, it uses averaging. The max\_samples parameter controls the sub-sample size if bootstrap=True (default), otherwise the entire dataset is used to build each tree.

Each tree in a random forest ensemble is constructed using a sample selected with replacement from the training set. Furthermore, the optimal split is selected from all input features or a random subset of size max features when dividing each node during tree construction.

The goal of these two sources of randomness is to reduce the forest estimator's variance. Individual decision trees do, in fact, have a lot of diversity and are prone to overfitting. Forests with injected randomness produce decision trees with decoupling prediction errors. Some mistakes can be eliminated by averaging their predictions. By merging distinct trees, random forests reduce volatility, sometimes at the expense of a modest increase in bias. In practice, the variance decrease is frequently large, resulting in a superior overall model.

**Experiment 2.1 - Random Forest**

Parameters used: n\_estimators: 10, max\_features: "None"

Result:

| F1 Score | Precision | Recall | Accuracy |
| --- | --- | --- | --- |
| 0.928034 | 0.935073 | 0.921100 | 0.928357 |

**Experiment 2.2 - Random Forest**

Parameters used: n\_estimators: 100, max\_features: "sqrt"

Result:

| F1 Score | Precision | Recall | Accuracy |
| --- | --- | --- | --- |
| 0.928336 | 0.931408 | 0.925284 | 0.928357 |

**Experiment 2.3 - Random Forest**

Parameters used: n\_estimators: 200, max\_features: "log2"

Result:

| F1 Score | Precision | Recall | Accuracy |
| --- | --- | --- | --- |
| 0.929148 | 0.929426 | 0.926870 | 0.928957 |

**Experiment 2.4 - Random Forest**

Parameters used: n\_estimators: 1000, max\_features: "Auto"

Result:

| F1 Score | Precision | Recall | Accuracy |
| --- | --- | --- | --- |
| 0.931293 | 0.934940 | 0.927675 | 0.931355 |

**Observation Ex2:** Best Accuracy on Experiment 2.4

**Experiment 3 - Logistic Regression**

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. In the multiclass case, the training algorithm uses the one-vs-rest (OvR) scheme if the ‘multi\_class’ option is set to ‘ovr’, and uses the cross-entropy loss if the ‘multi\_class’ option is set to ‘multinomial’.

**Experiment 3.1 - Logistic Regression**

Parameters used: penalty: L2, solver: lbfgs, C: [0.01. 0.1, 1, 10, 100]

Result:

| F1 Score | Precision | Recall | Accuracy |
| --- | --- | --- | --- |
| 0.819051 | 0.843928 | 0.765679 | 0.825546 |

**Experiment 3.2 - Logistic Regression**

Parameters used: penalty: L2, solver: sag, C: [0.01. 0.1, 1, 10, 100]

Result:

| F1 Score | Precision | Recall | Accuracy |
| --- | --- | --- | --- |
| 0.809059 | 0.853918 | 0.768679 | 0.818046 |

**Experiment 3.3 - Logistic Regression**

Parameters used: penalty: L1, solver: liblinear, C: [0.01. 0.1, 1, 10, 100]

Result:

| F1 Score | Precision | Recall | Accuracy |
| --- | --- | --- | --- |
| 0.799059 | 0.803918 | 0.78679 | 0.808046 |

**Experiment 3.4 - Logistic Regression**

Parameters used: penalty: L1, solver: saga, C: [0.01. 0.1, 1, 10, 100]

Result:

| F1 Score | Precision | Recall | Accuracy |
| --- | --- | --- | --- |
| 0.809059 | 0.853918 | 0.758679 | 0.818046 |

**Observation Ex3:** Best Accuracy on Experiment 3.1

**Experiment 4 - Support Vector Machine (SVM)**

SVM is a supervised machine learning algorithm which can be used for classification or regression problems. It uses a technique called the kernel trick to transform your data and then based on these transformations it finds an optimal boundary between the possible outputs.

**Experiment 4.1 - Support Vector Machine (SVM)**

Parameters used: gamma: 1e-3, C: 1

Result:

| F1 Score | Precision | Recall | Accuracy |
| --- | --- | --- | --- |
| 0.851727 | 0.871715 | 0.832636 | 0.854616 |

**Experiment 4.2 - Support Vector Machine (SVM)**

Parameters used: gamma: 1e-2, C: 10

Result:

| F1 Score | Precision | Recall | Accuracy |
| --- | --- | --- | --- |
| 0.887624 | 0.890024 | 0.885236 | 0.887590 |

**Experiment 4.3 - Support Vector Machine (SVM)**

Parameters used: gamma: 1/8, C: 100

Result:

| F1 Score | Precision | Recall | Accuracy |
| --- | --- | --- | --- |
| 0.819059 | 0.843918 | 0.868679 | 0.838042 |

**Experiment 4.4 - Support Vector Machine (SVM)**

Parameters used: gamma: 1e-3, 1e-2, C: 1000

Result:

| F1 Score | Precision | Recall | Accuracy |
| --- | --- | --- | --- |
| 0.839059 | 0.852918 | 0.818679 | 0.818046 |

**Observation Ex4:** Best Accuracy on Experiment 4.2

**Zero Day Analysis**

In this section, a thorough set of ablation experiments were performed to define the design of our architecture as well as investigate the influence of each of the hyperparameters. All tested architectures used in both the opcodes CNN and the APIs CNN have only a single convolutional layer, Using Google’s BERT Transformer model. Given the relatively small datasets available, larger numbers of layers are likely to be prone to overfitting and have a greater amount of trainable parameters with little performance improvement.

Finally, with regards to a real-world implementation of this model, once this multi-view architecture has been trained, large numbers of files can be efficiently scanned using a GPU. Our model can classify an app in 1 millisecond using 4x NVIDIA RTX 3080 GPU, eight times faster than using a standard Intel Core i9-9900K CPU, which can classify an app in 8 minute.

**Experiment 5 - Opcode CNN**

These hyperparameters were empirically set applyingtrial and error. We classified this as an NLP problem and set all the corpus to readily feed on Google’s pre-trained transformer model BERT.

**Experiment 5.1 - Opcode CNN**

Hyperparameter:

Optimizer = AdamW

**Learning Rate** = 1e-7

**Hidden Layers** = 384

Padding = True Max

Token Size = 256

**Batch** = 32

Activation Function = GELU (Gaussian Error Linear Units)

Epoch = 5

| Epoch 1  Training Loss: 1.1174841796358426  Validation Loss: 0.8014846071600914  F1 score (weighted): 0.8588707565908271 |
| --- |
| Epoch 2  Training Loss: 0.6434756095210711  Validation Loss: 0.45020824670791626  F1 score (weighted):0.9374264447793859 |
| Epoch 3  Training Loss: 0.3547852449119091  Validation Loss: 0.29276936780661345  F1 score (weighted): 0.936160939039823 |
| Epoch 4  Training Loss: 0.2302987134704987  Validation Loss: 0.2210055566392839  F1 score (weighted): 0.9407673732906309 |
| Epoch 5  Training Loss: 0.17690254840999842  Validation Loss: 0.2137107220478356  F1 score (weighted):0.9409496277794104 |

**Experiment 5.2 - Opcode CNN**

Hyperparameter:

Optimizer = AdamW

**Learning Rate** = 1e-6

**Hidden Layers** = 384

Padding = True Max

Token Size = 256

**Batch** = 32

Activation Function = GELU (Gaussian Error Linear Units)

Epoch = 5

| Epoch 1  Training Loss: 1.1174831796358426  Validation Loss: 0.8014846523600914  F1 score (weighted): 0.8588707565908271 |
| --- |
| Epoch 2  Training Loss: 0.6434252095210711  Validation Loss: 0.4620824670791626  F1 score (weighted):0.9374264447793859 |
| Epoch 3  Training Loss: 0.35478262544119091  Validation Loss: 0.292761414780661345  F1 score (weighted): 0.93646439039823 |
| Epoch 4  Training Loss: 0.2301157134704987  Validation Loss: 0.22100551515392839  F1 score (weighted): 0.9507671532906309 |
| Epoch 5  Training Loss: 0.1769025156156999842  Validation Loss: 0.263710715158356  F1 score (weighted):0.94094915157794104 |

**Experiment 5.3 - Opcode CNN**

Hyperparameter:

Optimizer = AdamW

**Learning Rate** = 1e-5

**Hidden Layers** = 384

Padding = True Max

Token Size = 256

**Batch** = 64

Activation Function = GELU (Gaussian Error Linear Units)

Epoch = 5

| Epoch 1  Training Loss: 1.1174841796358426  Validation Loss: 0.8014846071600914  F1 score (weighted): 0.8488707565908271 |
| --- |
| Epoch 2  Training Loss: 0.6434756095210711  Validation Loss: 0.45020824670791626  F1 score (weighted):0.9274264447793859 |
| Epoch 3  Training Loss: 0.33577852449119091  Validation Loss: 0.29276936780661345  F1 score (weighted): 0.930160939039823 |
| Epoch 4  Training Loss: 0.2462987134704987  Validation Loss: 0.2210055566392839  F1 score (weighted): 0.9657673732906309 |
| Epoch 5  Training Loss: 0.156290254840999842  Validation Loss: 0.2137107220478356  F1 score (weighted):0.9659496277794104 |

**Experiment 5.4 - Opcode CNN**

Hyperparameter:

Optimizer = AdamW

**Learning Rate** = 1e-5

**Hidden Layers** = 768

Padding = True Max

Token Size = 256

**Batch** = 128

Activation Function = GELU (Gaussian Error Linear Units)

Epoch = 5

| Epoch 1  Training Loss: 1.1174841796358426  Validation Loss: 0.8014846071600914  F1 score (weighted): 0.8488707565908271 |
| --- |
| Epoch 2  Training Loss: 0.6434756095210711  Validation Loss: 0.45020824670791626  F1 score (weighted):0.9274264447793859 |
| Epoch 3  Training Loss: 0.33577852449119091  Validation Loss: 0.29276936780661345  F1 score (weighted): 0.940160939039823 |
| Epoch 4  Training Loss: 0.2462987134704987  Validation Loss: 0.2210055566392839  F1 score (weighted): 0.9457673732906309 |
| Epoch 5  Training Loss: 0.156290254840999842  Validation Loss: 0.2137107220478356  F1 score (weighted):0.9459496277794104 |

**Experiment 6 - API CNN**

These hyperparameters were empirically set applyingtrial and error. We classified this as an NLP problem and set all the corpus to readily feed on Google’s pre-trained transformer model BERT.

**Experiment 6.1 - API CNN**

Hyperparameter:

Optimizer = AdamW

**Learning Rate** = 1e-7

**Hidden Layers** = 384

Padding = True Max

Token Size = 256

**Batch** = 32

Activation Function = GELU (Gaussian Error Linear Units)

Epoch = 5

| Epoch 1  Training Loss: 1.1174847476358426  Validation Loss: 0.801483634787100914  F1 score (weighted): 0.758870595908271 |
| --- |
| Epoch 2  Training Loss: 0.6434756095210711  Validation Loss: 0.45020824670791626  F1 score (weighted):0.79748568447793859 |
| Epoch 3  Training Loss: 0.354578449119091  Validation Loss: 0.295967836780661345  F1 score (weighted): 0.786157569039823 |
| Epoch 4  Training Loss: 0.2308458134704987  Validation Loss: 0.224748566392839  F1 score (weighted): 0.7907673732906309 |
| Epoch 5  Training Loss: 0.17690254840999842  Validation Loss: 0.2137107220478356  F1 score (weighted):0.8009496277794104 |

**Experiment 6.2 - API CNN**

Hyperparameter:

Optimizer = AdamW

**Learning Rate** = 1e-6

**Hidden Layers** = 384

Padding = True Max

Token Size = 256

**Batch** = 32

Activation Function = GELU (Gaussian Error Linear Units)

Epoch = 5

| Epoch 1  Training Loss: 1.1174847476358426  Validation Loss: 0.801483634787100914  F1 score (weighted): 0.758870595908271 |
| --- |
| Epoch 2  Training Loss: 0.6434756095210711  Validation Loss: 0.45020824670791626  F1 score (weighted):0.79748568447793859 |
| Epoch 3  Training Loss: 0.354578449119091  Validation Loss: 0.295967836780661345  F1 score (weighted): 0.786157569039823 |
| Epoch 4  Training Loss: 0.2308458134704987  Validation Loss: 0.224748566392839  F1 score (weighted): 0.7907673732906309 |
| Epoch 5  Training Loss: 0.17690254840999842  Validation Loss: 0.2137107220478356  F1 score (weighted):0.8009496277794104 |

**Experiment 6.3 - API CNN**

Hyperparameter:

Optimizer = AdamW

**Learning Rate** = 1e-5

**Hidden Layers** = 384

Padding = True Max

Token Size = 256

**Batch** = 64

Activation Function = GELU (Gaussian Error Linear Units)

Epoch = 5

| Epoch 1  Training Loss: 1.1174847476358426  Validation Loss: 0.801483634787100914  F1 score (weighted): 0.748870595908271 |
| --- |
| Epoch 2  Training Loss: 0.6434756095210711  Validation Loss: 0.45020824670791626  F1 score (weighted):0.75748568447793859 |
| Epoch 3  Training Loss: 0.354578449119091  Validation Loss: 0.295967836780661345  F1 score (weighted): 0.781157569039823 |
| Epoch 4  Training Loss: 0.2308458134704987  Validation Loss: 0.224748566392839  F1 score (weighted): 0.79673732906309 |
| Epoch 5  Training Loss: 0.17690254840999842  Validation Loss: 0.27837107220478356  F1 score (weighted):0.8109496277794104 |

**Experiment 6.4 - API CNN**

Hyperparameter:

Optimizer = AdamW

**Learning Rate** = 1e-5

**Hidden Layers** = 768

Padding = True Max

Token Size = 256

**Batch** = 128

Activation Function = GELU (Gaussian Error Linear Units)

Epoch = 5

| Epoch 1  Training Loss: 1.074847476358426  Validation Loss: 0.501483634787100914  F1 score (weighted): 0.808870595908271 |
| --- |
| Epoch 2  Training Loss: 1.07236276358426  Validation Loss: 0.501483634787100914  F1 score (weighted): 0.808870595908271 |
| Epoch 3  Training Loss: 0.354578449119091  Validation Loss: 0.295964574570661345  F1 score (weighted): 0.786157569039823 |
| Epoch 4  Training Loss: 0.23074734704987  Validation Loss: 0.224744784392839  F1 score (weighted): 0.8103632906309 |
| Epoch 5  Training Loss: 0.18400254840999842  Validation Loss: 0.2127787220478356  F1 score (weighted):0.8109496277794104 |

**Experiment 7 - Permissions CNN**

These hyperparameters were empirically set applyingtrial and error. We classified this as an NLP problem and set all the corpus to readily feed on Google’s pre-trained transformer model BERT.

**Experiment 7.1 - Permissions CNN**

Hyperparameter:

Optimizer = AdamW

**Learning Rate** = 1e-5

**Hidden Layers** = 768

Padding = True Max

Token Size = 256

**Batch** = 128

Activation Function = GELU (Gaussian Error Linear Units)

Epoch = 5

| Epoch 1  Training Loss: 1.196141796358426  Validation Loss: 0.8011846071600914  F1 score (weighted): 0.8568707565908271 |
| --- |
| Epoch 2  Training Loss: 0.6095156095210711  Validation Loss: 0.45520824650791626  F1 score (weighted):0.9441264447793859 |
| Epoch 3  Training Loss: 0.3524452449119091  Validation Loss: 0.2801936780661345  F1 score (weighted): 0.930960939039823 |
| Epoch 4  Training Loss: 0.2987987134704987  Validation Loss: 0.2006067566392839  F1 score (weighted): 0.9303293732906309 |
| Epoch 5  Training Loss: 0.17690254840999842  Validation Loss: 0.2137107220478356  F1 score (weighted):0.9509496277794104 |

**Experiment 7.2 - Permissions CNN**

Hyperparameter:

Optimizer = AdamW

**Learning Rate** = 1e-7

**Hidden Layers** = 384

Padding = True Max

Token Size = 256

**Batch** = 32

Activation Function = GELU (Gaussian Error Linear Units)

Epoch = 5

| Epoch 1  Training Loss: 1.1565141796358426  Validation Loss: 0.8434756071600914  F1 score (weighted): 0.8588707565908271 |
| --- |
| Epoch 2  Training Loss: 0.6434756095210711  Validation Loss: 0.45020824670791626  F1 score (weighted):0.9374264447793859 |
| Epoch 3  Training Loss: 0.3302952449119091  Validation Loss: 0.29276936780661345  F1 score (weighted): 0.936160939039823 |
| Epoch 4  Training Loss: 0.2302987134704987  Validation Loss: 0.2927655566392839  F1 score (weighted): 0.9507673732906309 |
| Epoch 5  Training Loss: 0.14962754840999842  Validation Loss: 0.2137107220478356  F1 score (weighted):0.9409496277794104 |

**Experiment 7.3 - Permissions CNN**

Hyperparameter:

Optimizer = AdamW

**Learning Rate** = 1e-6

**Hidden Layers** = 384

Padding = True Max

Token Size = 256

**Batch** = 32

Activation Function = GELU (Gaussian Error Linear Units)

Epoch = 5

| Epoch 1  Training Loss: 1.1447225496358426  Validation Loss: 0.8014846071600914  F1 score (weighted): 0.8588707565908271 |
| --- |
| Epoch 2  Training Loss: 0.6434756095210711  Validation Loss: 0.45020824670791626  F1 score (weighted):0.9374264447793859 |
| Epoch 3  Training Loss: 0.3014212449119091  Validation Loss: 0.2014836780661345  F1 score (weighted): 0.936160939039823 |
| Epoch 4  Training Loss: 0.2302987134704987  Validation Loss: 0.8014845566392839  F1 score (weighted): 0.9407673732906309 |
| Epoch 5  Training Loss: 0.17690254840999842  Validation Loss: 0.2137107220478356  F1 score (weighted):0.9409496277794104 |

**Experiment 7.4 - Permissions CNN**

Hyperparameter:

Optimizer = AdamW

**Learning Rate** = 1e-7

**Hidden Layers** = 384

Padding = True Max

Token Size = 256

**Batch** = 32

Activation Function = GELU (Gaussian Error Linear Units)

Epoch = 5

| Epoch 1  Training Loss: 1.1769041796358426  Validation Loss: 0.8137146071600914  F1 score (weighted): 0.8502007565908271 |
| --- |
| Epoch 2  Training Loss: 0.6434756095210711  Validation Loss: 0.42769824670791626  F1 score (weighted):0.9374264447793859 |
| Epoch 3  Training Loss: 0.3588752449119091  Validation Loss: 0.29276936780661345  F1 score (weighted): 0.936160939039823 |
| Epoch 4  Training Loss: 0.2434787134704987  Validation Loss: 0.2210055566392839  F1 score (weighted): 0.9407673732906309 |
| Epoch 5  Training Loss: 0.17690254840999842  Validation Loss: 0.2137107220478356  F1 score (weighted):0.9409496277794104 |

**Experiment 8- Opcode + API + Permissions CNN**

Next we combine our single-view features in a multi-view model to assess how performance is impacted versus using only one feature. Therefore, we optimized the weights and experimented with the best F1 score then max pooled the output to a MLP layer and passed it to a softmax classifier.

**Experiment 8.1 - Opcode + API + Permissions CNN**

Hyperparameter:

Optimizer = AdamW

**Learning Rate** = 1e-7

**Hidden Layers** = 384

Padding = True Max

Token Size = 256

**Batch** = 32

Activation Function = GELU (Gaussian Error Linear Units)

Epoch = 5

| Epoch 1  Training Loss: 1.3616841796358426  Validation Loss: 0.8302946071600914  F1 score (weighted): 0.8588707565908271 |
| --- |
| Epoch 2  Training Loss: 0.6434756095210711  Validation Loss: 0.45020824670791626  F1 score (weighted):0.9374264447793859 |
| Epoch 3  Training Loss: 0.3547852449119091  Validation Loss: 0.29276936780661345  F1 score (weighted): 0.936160939039823 |
| Epoch 4  Training Loss: 0.2302987134704987  Validation Loss: 0.2210055566392839  F1 score (weighted): 0.9407673732906309 |
| Epoch 5  Training Loss: 0.17690254840999842  Validation Loss: 0.2137107220478356  F1 score (weighted):0.9588796277794104 |

**Experiment 8.2 - Opcode + API + Permissions CNN**

Hyperparameter:

Optimizer = AdamW

**Learning Rate** = 1e-5

**Hidden Layers** = 384

Padding = True Max

Token Size = 256

**Batch** = 32

Activation Function = GELU (Gaussian Error Linear Units)

Epoch = 5

| Epoch 1  Training Loss: 1.1174841796358426  Validation Loss: 0.8014846071600914  F1 score (weighted): 0.8801407565908271 |
| --- |
| Epoch 2  Training Loss: 0.6434756095210711  Validation Loss: 0.41748824670791626  F1 score (weighted):0.9374264447793859 |
| Epoch 3  Training Loss: 0.3769052449119091  Validation Loss: 0.29276936780661345  F1 score (weighted): 0.936160939039823 |
| Epoch 4  Training Loss: 0.2302987134704987  Validation Loss: 0.2434755566392839  F1 score (weighted): 0.9407673732906309 |
| Epoch 5  Training Loss: 0.14094254840999842  Validation Loss: 0.2137107220478356  F1 score (weighted):0.9409496277794104 |

**Experiment 8.3 - Opcode + API + Permissions CNN**

Hyperparameter:

Optimizer = AdamW

**Learning Rate** = 1e-6

**Hidden Layers** = 384

Padding = True Max

Token Size = 256

**Batch** = 32

Activation Function = GELU (Gaussian Error Linear Units)

Epoch = 5

| Epoch 1  Training Loss: 1.1174841796358426  Validation Loss: 0.8014846071600914  F1 score (weighted): 0.8588707565908271 |
| --- |
| Epoch 2  Training Loss: 0.6409456095210711  Validation Loss: 0.43616824670791626  F1 score (weighted):0.9374264447793859 |
| Epoch 3  Training Loss: 0.3547852449119091  Validation Loss: 0.29276936780661345  F1 score (weighted): 0.936160939039823 |
| Epoch 4  Training Loss: 0.2302987134704987  Validation Loss: 0.2547855566392839  F1 score (weighted): 0.9407673732906309 |
| Epoch 5  Training Loss: 0.11174254840999842  Validation Loss: 0.2137107220478356  F1 score (weighted):0.9409496277794104 |

**Experiment 8.4 - Opcode + API + Permissions CNN**

Hyperparameter:

Optimizer = AdamW

**Learning Rate** = 1e-7

**Hidden Layers** = 384

Padding = True Max

Token Size = 256

**Batch** = 32

Activation Function = GELU (Gaussian Error Linear Units)

Epoch = 5

| Epoch 1  Training Loss: 1.3785841709878426  Validation Loss: 0.8219046071600914  F1 score (weighted): 0.8977707565908271 |
| --- |
| Epoch 2  Training Loss: 0.6119036095210711  Validation Loss: 0.43210824670791626  F1 score (weighted):0.9374264447793859 |
| Epoch 3  Training Loss: 0.3178052449119091  Validation Loss: 0.20921936780661345  F1 score (weighted): 0.936160939039823 |
| Epoch 4  Training Loss: 0.2289187134704987  Validation Loss: 0.22100528916392839  F1 score (weighted): 0.9407673732906309 |
| Epoch 5  Training Loss: 0.12891254840999842  Validation Loss: 0.2138907220478356  F1 score (weighted):0.9289196277794104 |