

**Artificial Neural Networks**

**Phishing Website Detection**

|  |  |  |
| --- | --- | --- |
| **Name** | **ID** | **NN** |
| **Hagar** | **206002** | **MLP** |
| **Yossuf** | **197032** | **SOM** |
| **Yasmine\*** | **206197** | **RBF** |

**Introduction:**

Phishing arises when online trading occurring with unsecured money orders. The term phishing is used as a form of internet crime that is the creation of a well-known website that is essentially bogus to obtain confidential information. To detect phishing websites one must identify the length of the URL, IP address, domain’s prefix and suffix. So, this dataset is aimed to create a model that predicts the phishing websites and the legitimate websites, capturing the most important features. The aim of our project is to get the best model to detect such dataset. The dataset does include duplicates, around 50% of the data, however none were dropped to help in training and generalizing the neural network models, furthermore most of the dataset is duplicated. Similarly, label encoding was used to normalize the data and to convert all the data from objects to integers. Standardization was not used on this dataset as it’s values range from -1 to 1, which is somewhat normalized. In addition to that, the dataset used is somewhat balanced by a ratio of 45:55, nevertheless balancing was used.

1. **Data Preparation/Clustering:**

* For outlier detections, we compared 3 methods to get the least number of outliers to drop as we need to retain as much data to not lose important features.

1. Elliptic Envelope (Gaussian-based Outlier Detection) results to 1k-4k outliers being detected.
2. Local Outlier Factor (KNN-Clustering) results to 500-2k outliers being detected.
3. Isolation Forest results to 4k outliers being detected.

* We ended up using the Local Outlier Factor (KNN-clustering) to detect the outliers to be
* For choose the best method to balance the data to be used in each model, we compared 3 methods.

1. Oversampling: Duplicates dataset’s minority class instances.
2. Undersampling: Removes dataset’s majority class instances.
3. SMOTE: Generates new samples from the minority class instances.

🡪 SMOTE is the best method to balance the dataset in accuracy and precision, as it generates new samples from the minority class rather than duplicating the minority class instances (oversampling) or lowering the majority class samples (undersampling).

1. **MLP models comparison:**

The Multi-Layer Perceptron (MLP) are made from neurons stacked in layers. Each neuron must be connected to other neurons on the next layer (especially between the hidden layers). In other words, the MLP neural network architecture is a fully connected neural network, so the dense class would be used. Data would traverse from the input layer to the next until the output layer is reached, through the layers in one direction, so sequential class is also needed. This also implies that MLPs are sequential.

1. **Simple MLPs – 2 hidden layers**

|  |  |  |
| --- | --- | --- |
| **Units** | **Same number of units (4, 4)** | **Different number of units (6, 4)** |
| **ROC Curve** |  |  |
| **Accuracy** | 95.2% | 95.6% |
| **Precision** | 95% | 96% |
| **Recall** | 95% | 96% |
| **Learning Curve** |  |  |
| **#Misclassifications** | 166 | 153 |

* The ROC curves show the area under the curve as the performance of both models. However, the best model by 0.4% according to its performance is the different-unit model (the one on the right).
* To further prove the point made above, the right-hand model has higher accuracy, precision (specificity) and recall (sensitivity) of 96%, difference of 1%. Meaning that the right-hand model is the better classifier to the point that it can predict correctly and relevantly by 96%.
* Despite the models here being similar (1% off), the number of misclassifications made by the models differ, the right-hand model has less misclassifications than the left-hand model.
* The models seem to be somewhat ideal, as both of their learning curves display a slight overfitting to the training set. Having their variance high, to the point where the training set is so well fit that it is sensitive to change.
* The optimal epoch for the highest accuracy and precision=100, 250 was also tried but lead to satisfactory results, the models became overfitted.
* Based solely on the number of misclassifications, the current best model is the right-hand model, predicting less misclassifications.

1. **Complex MLPs-4 hidden layers**

|  |  |  |
| --- | --- | --- |
| **Units** | **Same number of units (6,6,6,6)** | **Decreasing number of units (8,6,4,2)** |
| **ROC Curve** |  |  |
| **Accuracy** | 95.0% | 95.7% |
| **Precision** | 95% | 96% |
| **Recall** | 95% | 96% |
| **Learning Curve** |  |  |
| **#Misclassifications** | 174 | 149 |

* After adding 2 more hidden layers, the new models are similar to the ones who originally have 2 hidden layers in terms of accuracy, recall and precision. Nonetheless, the right-hand model outperforms the others by 0.01% at least.
* The ROC curves show the area under the curve as the performance of both models. However, the best model by 0.7% according to its performance is the different-unit model (the one on the right).
* To further prove the point made above, the right-hand model has higher accuracy, precision (specificity) and recall (sensitivity) of 95.7%, difference of 0.7%. Meaning that the right-hand model is the better classifier to the point that it can predict correctly and relevantly by 95.7%.
* Despite the models here being similar (0.7% off), the number of misclassifications made by the models differ, the right-hand model has less misclassifications than the left-hand model.
* The right-hand model seems to be overfitting as the training errors are lower than the testing errors by 0.6. This high variation can mean that the data is taught to for so well that it is sensitive to change.
* Based solely on the number of misclassifications, the current best model is the right-hand model, predicting less misclassifications.

|  |  |  |
| --- | --- | --- |
| **Activation Function** | **Best model = 4 hidden layers (mainly RELU)- Decreasing number of units (8,6,4,2)** | **Best 4 hidden layers (mainly Tanh)** |
| **ROC Curve** |  |  |
| **Accuracy** | 95.7% | 95.3% |
| **Precision** | 96% | 95% |
| **Recall** | 96% | 95% |
| **Learning Curve** |  |  |
| **#Misclassifications** | 149 | 161 |

* By changing the activation function from RELU to tanh, the tanh-based model is kind of inaccurate and imprecise relative to the RELU-based model by 0.4% in accuracy and 1% in precision.
* The recall (sensitivity) of the models are 1% off each other, so the rate of the true positive sample classified by the RELU-based model is higher than the tanh-based model.
* Both learning curves show that both models have overfitted the training dataset since their training only learns from its data, so when they get tested, they have higher testing errors (in relation to their training errors).
* Also the number of misclassifications made in the RELU-based model is less, thus making it still the best model.

|  |  |  |
| --- | --- | --- |
| **Loss Functions** | **Best model = 4 hidden layers (mainly RELU)- Decreasing number of units (8,6,4,2)- binary\_crossentropy** | **4 hidden layers (mainly RELU)- Decreasing number of units (8,6,4,2)- hinge** |
| **ROC Curve** |  |  |
| **Accuracy** | 95.7% | 95.2% |
| **Precision** | 96% | 95% |
| **Recall** | 96% | 95% |
| **Learning Curve** |  |  |
| **#Misclassifications** | 149 | 165 |

* So in closing, the best MLP model (after compare with many other models) is the 4 hidden layers, mainly RELU as its activation function when varying number of units that uses binary cross entropy to classify the dataset.
* This shows that regardless the number of hidden layers and different numbers of units, MLP models depend on for performance in such dataset is the loss function.
* As it outperforms the hinge-based model in accuracy (0.5%), precision (1%) and recall (1%). However, it clearly overfits the dataset, as the variance is higher in it relative to the hinge-based model. This means that it fits too well with the training data, it overacts to minor changes.
* On the other hand, the only model that did not overfit the data is the hinge-based model.
* Nevertheless, it has a fewer number of misclassifications than the hinge-based model.

1. **SOM Models Comparisons:**

SUSI: Supervised Self-Organizing Maps For Regression And Classification

To apply the SuSi framework for solving supervised classification tasks, SUSI differs from unsupervised SOM in the dimension of the weights and their estimation algorithm. The weights of the unsupervised SOM have the same dimension as the input data. Thus, adapting these weights often changes the BMU for each input data point. In contrast, the weights of the supervised SOM have the same dimension as the target variable of the respective task. One has to distinguish between two cases: regression and classification (SUSI) can be used in regression too. In the regression case, the weights are one-dimensional and contain a continuous number. In the classification case, the weights contain a class. By combining the unsupervised and the supervised SOM, the former is used to select the BMU for each data point while the latter links the selected BMU to a specific estimation. In the following.

**SOM default parameters:**

• Which are n\_rows: int = 10,

• n\_columns: int = 10,

• init\_mode\_unsupervised: str = 'random',

• init\_mode\_supervised: str = 'majority',

• n\_iter\_unsupervised: int = 1000,

• n\_iter\_supervised: int = 1000,

**Best parameters with “Phishing Websites dataset”**

• Which are n\_rows: int = 40,

• n\_columns: int = 40,

• init\_mode\_unsupervised: str = 'random',

• init\_mode\_supervised: str = 'majority',

• n\_iter\_unsupervised: int = 10,

• n\_iter\_supervised: int = 10,

|  |  |  |
| --- | --- | --- |
| **Metrics** | **With default parameters** | With the best parameters |
| **Accuracy** | 58.5 % | 67.2 % |
| **Confusion matrix** |  |  |
| **precision** | 0.59 | 0.67 |
| **recall** | 0.62 | 0.67 |
| **f1-score** | 0.59 | 0.67 |
| **Roc curve** |  |  |
| **Learning curves** |  |  |

Comment: after increasing the size of the map and decreased number of iteration by a lot; the model performance got better results in all Metrics compare to the default one, regarding learning curves; there aren't over fitting or under fitting since the variance between training and test sets wasn't large in both of them.

1. **Overall Best Model Comparison:**

|  |  |  |
| --- | --- | --- |
| **Optimized Model** | **MLP-based** | **SOM-based** |
| **ROC Curve** |  |  |
| **Accuracy** | 95.7% | 67.2 % |
| **Precision** | 96% | 67% |
| **Recall** | 96% | 67% |
| **F1-score** | 96% | 67% |
| **Learning Curve** |  |  |
| **#Misclassifications** | 149 | 2554 |

* The best model to predict whether a website is a phishing (illegitimate) website: MLP-based model (according to its performance in the area under the curve in the ROC curve, making it the best classifier (predictor).
* the best MLP model (after compare with many other models) is the 4 hidden layers, mainly RELU as its activation function when varying number of units that uses binary cross entropy to classify the dataset.
* Binary cross entropy is used as a loss function for classification tasks.
* To further prove the point made above, the MLP model has higher accuracy, precision (specificity) and recall (sensitivity) of 95.7%, difference of more than 20%. Meaning that the MLP model is the better classifier to the point that it can predict correctly and relevantly by 95.7%.
* The ROC curves show the area under the curve (AUC) as the performance of the models. The MLP model is the best classifier as its area under the curve is almost 1, not like the other models, which have the AUC being less than 0.8.
* In addition to that, the MLP model predicts fewer mistakes than its counterpart. As it only misclassifies 149/11000 instances, while the SOM model misclassifies 2554/11000 samples.
* Despite the MLP model’s performance, its learning curve shows that it is overfitting (making it have a higher variance), as it learns the training data so well that it has less training errors than its testing data. While, the SOM model is ideal, as it does not underfit or overfit, in spite of having a poor performance relative to the MLP model.