Project report on

"Fake Instagram Profile Detection Model"

as a part of six-week summer Internship

on

Machine Learning and its application in Cyber Security

Under the mentorship of

Mrs. Nandini Sethi

Organized by

Department of IT

Indira Gandhi Delhi Technical University for Women

Patron

Dr (Mrs.) Amita Dev, Hon'ble Vice Chancellor, IGDTUW

Acknowledgement

I would like to express a deep sense of thanks & gratitude to my teacher Ms. Nandini Sethi and course instructor Dr. Santanoo Pattnaik for teaching and guiding me immensely through the duration of my course. I have been able to understand and grasp many new and interesting concepts of Machine Learning under his guidance. His constructive advice & constant motivation has been responsible for the successful completion of this project. I am deeply grateful to Honourable Vice Chancellor Dr. Amita Dev ma'am for providing me this wonderful opportunity which has helped me in growing and developing my skills.

Abstract

This project is about **"Fake Instagram Profile Detection"** that allows to identify whether the Instagram profile is genuine or fake. Algorithms are trained using all previous user's fake and authentic account data. When we provide new data, the trained model is applied to the new test data to determine whether the given new account details are from genuine or fake users. The machine learning algorithms are used to identify false accounts that could give wrong impressions about people. The dataset is pre-processed using variety of python libraries and a comparison form is generated to get a realistic algorithm suitable for the specified dataset. The classification algorithms **Random Forest** and **Support Vector Machine** are used for detection of fake accounts. This method can be applied easily by online social networks that have millions of profiles where profiles cannot be examined manually.

INDEX

Introduction

Social networking site is a website where each user has a profile and can keep in contact with friends, share their updates, meet new people who have the same interests. These Online Social Networks (OSN) use web2.0 technology, which allows users to interact with each other. Social networking sites are growing rapidly and changing the way people keep in contact with each other. The online communities bring people with the same interests together which makes users easier to make new friends. This is very important for marketing companies and celebrities who try to promote themselves by growing their base of followers and fans.

But there are a lot of issues too i.e., privacy, online bullying, misuse, trolling, etc. A lot of data is created and shared across the world. Many illegitimate users engage in fraudulent activities against social network users. Fake profiles can harm more than any other form of crime. This crime has to be detected. These fake accounts challenge the security of the system.

Many algorithms use huge volume of unstructured data generated from social medias. We aim to detect fake user profiles on Instagram. We are focusing on a smart predicting system to handle this problem based on prediction and classification techniques.

Objective

The main objective of our paper is identifying whether the Instagram profile is genuine or fake. Algorithms will be trained with all previous users fake and genuine account data and then whenever we give new test data then trained model will be applied on new test data to identify whether given new account details are from genuine or fake users. The machine learning-based methods were used to perceive false accounts that could give the wrong impression about people. The dataset is pre-processed using a variety of python libraries and a comparison form is obtained to get a realistic algorithm appropriate for the specified dataset . An effort to notice forged accounts on the social media platforms is strong-minded by a

variety of machine learning algorithms. The performances of the classification algorithms Random Forest, Support vector machines are used for the detection of fake accounts

Social impact

The Instagram craze allows brands to gain visibility through influencers who can showcase them. However, this opportunity also leads to a blight that distorts influencer marketing the purchase of likes and followers by some Instagram accounts. Phishing, a scam that generally is about someone trying to access your Instagram account by sending you a suspicious message or link that asks for your personal information. A model like this would not only safeguard an individual from the risks of cyber threats, brands are starting to be wary of these fake accounts that they no longer want to spend their influencer marketing budget on. attracted by the lure of financial gain and the desire to work with brands. Unfortunately, this totally breaks the very essence of influence: trust.

Proposed framework

The proposed framework in figure 1 shows the sequence of processes that need to be followed for continues detection of fake profiles with active learning from the feedback of the result given by the classification algorithm. This framework can easily be implemented by social networking companies.

1. The detection process starts with the selection of the profile that needs to be tested.

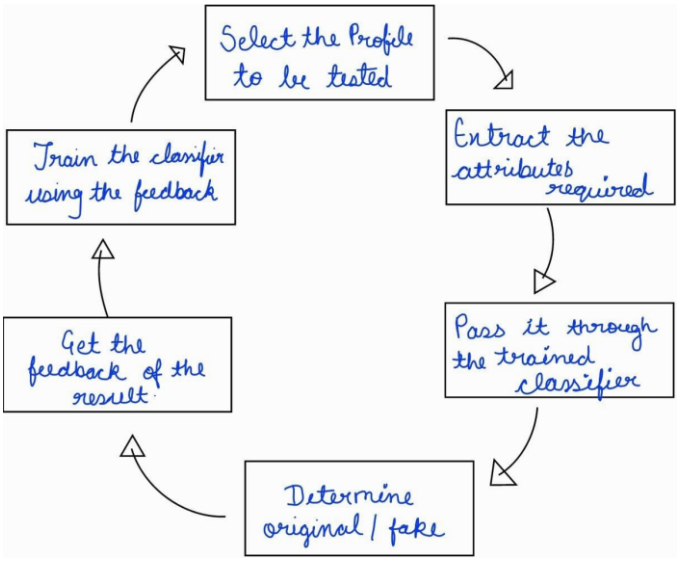
2. After the selection of the profile, the suitable attributes (i.e. features) are selected on which the classification algorithm is implemented.

3. The attributes extracted is passed to the trained classifier. The classifier gets trained regularly as new training data is feed into the classifier.

4. The classifier determines whether the profile is fake or genuine.

5. The classifier may not be 100% accurate in classifying the profile so; the feedback of the result is given back to the classifier.

6. This process repeats and as the time proceeds, the no. of training data increases and the classifier becomes more and more accurate in predicting the fake profiles.



Random Forest Tree

Random Forest is the machine learning algorithmic program that uses a textile approach to make a bunch of call trees with random set. A model is trained a lot random sample of the dataset to realize sane prediction performance many times. The output of all the choice trees within the tree, combined to create the ultimate prediction. For instance, within the higher than example - if five friends decide that you simply can like building R however solely a pair of friends decide that you simply won't just like the building then the ultimate prediction is that, you may like building R as majority continually wins.

Features of a Random Forest Algorithm

- It's more accurate than the decision tree algorithm.
- It provides an effective way of handling missing data.
- It can produce a reasonable prediction without hyper-parameter tuning.
- It solves the issue of overfitting in decision trees.

- In every random forest tree, a subset of features is selected randomly at the node's splitting point.

Working of Random Forest Algorithm

Random Forest works in two-phase first is to create the random forest by combining X decision tree, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps and diagram:

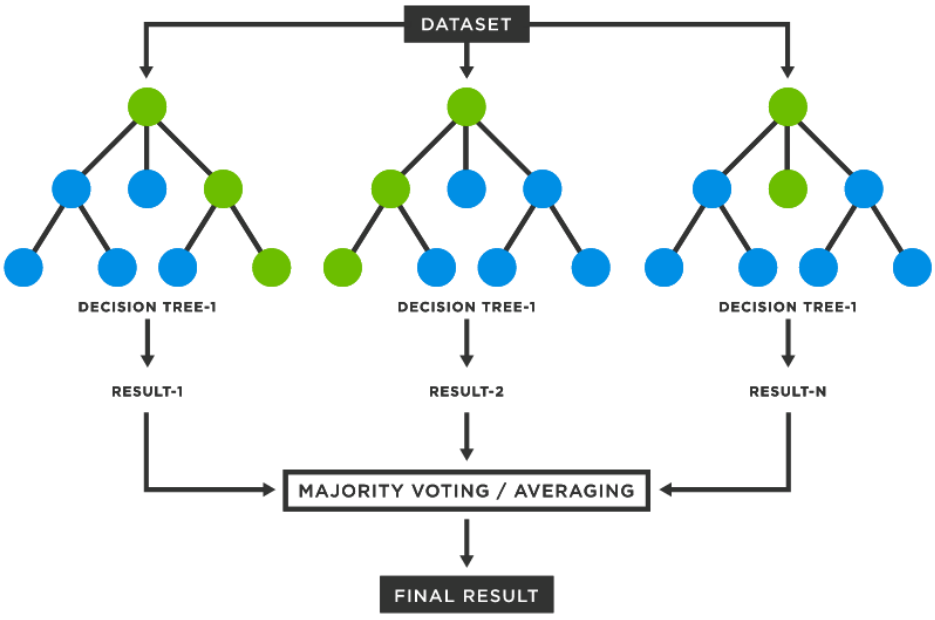
Step-1: Select random N data points from the training set.

Step-2: Build the decision trees associated with the selected data points (Subsets).

Step-3: Choose the number X for decision trees that you want to build.

Step-4: Repeat Step 1 & 2.

Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.



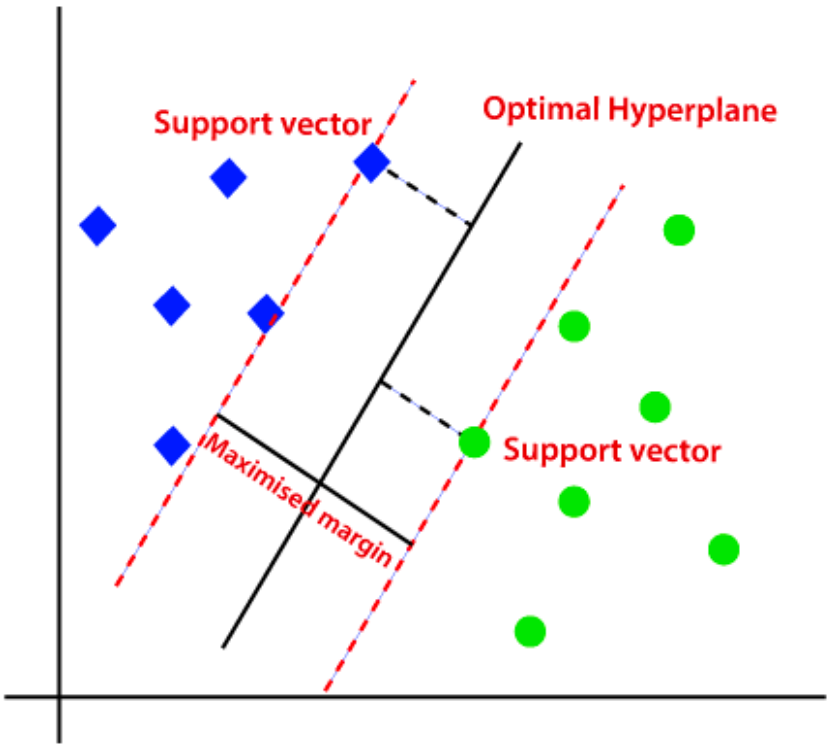
SVM algorithm

Machine learning involves predicting and classifying the data tends to use diverse these algorithms in line with the dataset. It will solve linear and non-linear issues and work well for several sensible issues. The thought of SVM is simple: The algorithmic program creates a hyper plane that separates the info into category. In machine learning, the radial basis operate kernel, could be a widespread kernel operate employed in varied kernelized learning algorithms. Especially, it's normally employed in support vector machine classification.

A simple linear SVM classifier works by making a straight line between two classes. That means all of the data points on one side of the line will represent a category and the data points on the other side of the line will be put into a different category. This means there can be an infinite number of lines to choose from.

What makes the linear SVM algorithm better than some of the other algorithms, like k-nearest neighbours, is that it chooses the best line to classify your data points. It chooses the line that separates the data and is the furthest away from the closet data points as possible.

A 2-D example helps to make sense of all the machine learning jargon. Basically you have some data points on a grid. You're trying to separate these data points by the category they should fit in, but you don't want to have any data in the wrong category. That means you're trying to find the line between the two closest points that keeps the other data points separated.



SVM can be understood with the example. Suppose we see dataset with both fake account and some real account, so if we want a model that can accurately classify whether it is a fake account or a real account, so such a model can be created by using the SVM algorithm.

We will first train our model with all previous user's fake and authentic account data so that it can learn about difference between fake and authentic account, and then we test it with our suspected account so as support vector creates a decision boundary between these two data and choose extreme cases (support vectors), it will see the extreme case of fake and authentic account. On the basis of the support vectors, it will classify it as a fake account.

Implementation

Data set

We needed a dataset of fake and genuine profiles. Various attributes included in the dataset are a number of friends, followers, status count. Dataset is divided into training and testing data. Classification algorithms are trained using a training dataset and the testing dataset is used to determine the efficiency of the algorithm. From the dataset used, 80% of both profiles (genuine and fake) are used to prepare a training dataset and 20% of both profiles are used to prepare a testing dataset.

Attribute

Table shows the Attributes considered for fake profile identification and the description for each of the attributes is provided.

S.No	Attribute	Description
1.	statuses_count	The number of views on the account.
2.	followers_count	The number of followers for the account.
3.	friends_count	The number of friends for the account.
4.	favourites_count	The number of likes on the recent post.
5.	listed_count	The number of close friends for the account.
6.	sex_code	The Gender of the account holder.
7.	lang_code	The language of account holder.

Evaluation Parameters (RF)

Efficiency/Accuracy = Number of predictions/Total Number of Predictions Percent Error = (1-Accuracy)*100

Confusion Matrix - Confusion Matrix is a technique for summarizing the performance of a classification algorithm. Calculating a confusion matrix can give you a better idea of what your classification model is getting right and what types of errors it is making.

TPR- True Positive Rate TPR=TP/(TP+FN)

FPR- False Positive Rate FPR=FP/(FP+TN)

TNR- True Negative Rate TNR=TN/(FP+TN)

FNR- False Negative Rate FNR=1-TPR

Recall- How many of the true positives were recalled (found), i.e. how many of the correct hits were also found.

Recall = TP / (TP+FN)

Precision- Precision is how many of the returned hits were true positive i.e. how many of the found were correct hits.

Precision = TP / (TP + FP)

F1 score- F1 score is a measure of a test's accuracy.

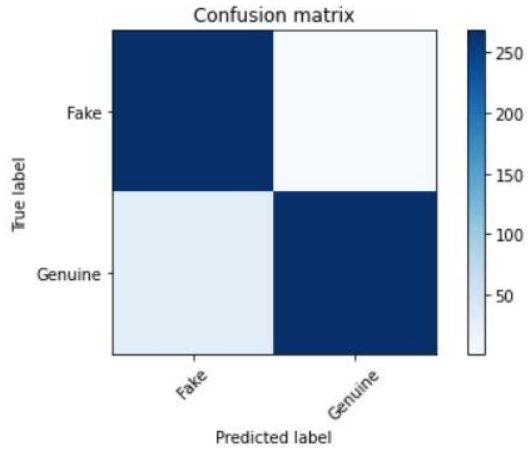
It considers both the precision p and the recall r of the test to compute the score.

ROC Curve- The Receiver Operating Characteristic is the plot of TPR versus FPR. ROC can be used to compare the performances of different classifiers.

Results

1. Confusion Matrix

```
Confusion matrix, without normalization
[[267   1]
 [ 28 268]]
```



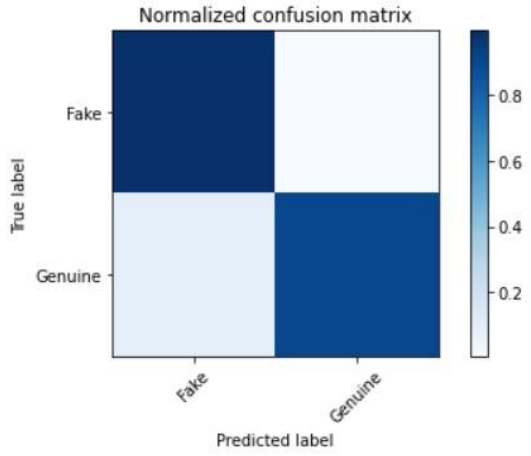
	Predicted FAKE	Predicted GENUINE
Actual FAKE	267	1
Actual GENUINE	28	268

Test Results as depicted by the Confusion Matrix

1. 267 occurrences when the model predicted a Fake account as Fake
2. 268 occurrences when the model predicted a Genuine account as Genuine
3. 1 occurrence when the model predicted a Fake account as Genuine
4. 28 occurrences when the model predicted a Genuine account as Fake

2. Normalised Confusion Matrix

```
Normalized confusion matrix
[[0.99626866 0.00373134]
 [0.09459459 0.90540541]]
```

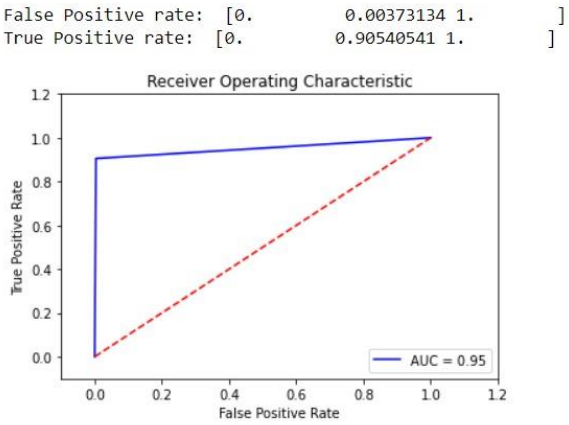


	Precision	recall	f1-score	support
Fake	0.91	1.00	0.95	268
Genuine	1.00	0.91	0.95	296
Accuracy			0.95	564
Macro avg	0.95	0.95	0.95	564
Weighted avg	0.95	0.95	0.95	564

Classification Report

The macro-averaged F1 score (or macro F1 score) is computed using the arithmetic mean (unweighted mean) of all the per-class F1 scores. This method treats all classes equally regardless of their support values.

3. ROC (Receiver Operating Characteristic



ROC curve

We plot the ROC (Receiver Operating Characteristic) curve to show the performance of a classification model at all thresholds. It plots two parameters:

1. True Positive Rate (0.00373134)
2. False Positive Rate (0.90540541)

The efficiency of the Random Forest Classifier in classifying data is **95%**. We have taken **80%** of the data for the training dataset and **20%** for the testing dataset.

Evaluation Parameters (SVM)

SVM is binary classification algorithm. Given a set of points of 2 types in N dimensional place, SVM generates a (N-1) dimensional hyper plane to separate those points into 2 groups. Say you have some points of 2 types in a paper which are linearly separable. SVM will find a straight line which separates those points into 2 types and situated as far as possible from all those points.

Generating confusion matrix and accuracy finding: Confusion Matrix is a technique for summarizing the performance of a classification algorithm. Calculating a confusion matrix can give you a better idea of what your classification model is getting right and what types of errors it is making.

True Positive Rate (T P R) = TP / TP + FN

False positive Rate (F P R) = FP / FP / + TN

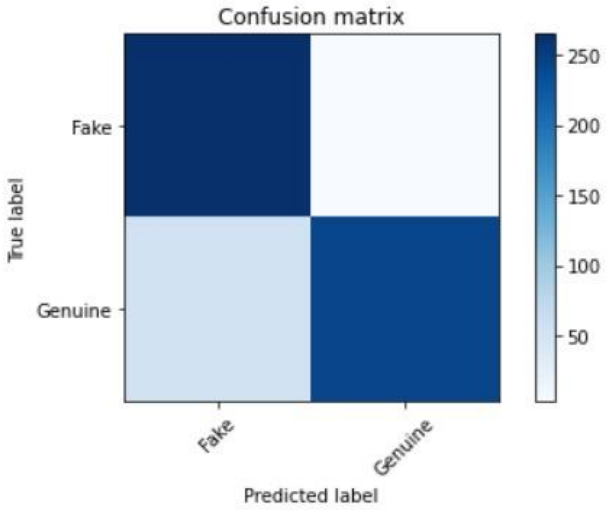
True Negative Rate (T N R) = TN/ FP + TN

False Negative Rate (F N R) = 1 –TPR

Results

1. Confusion Matrix

```
Confusion matrix, without normalization
[[265   3]
 [ 53 243]]
```



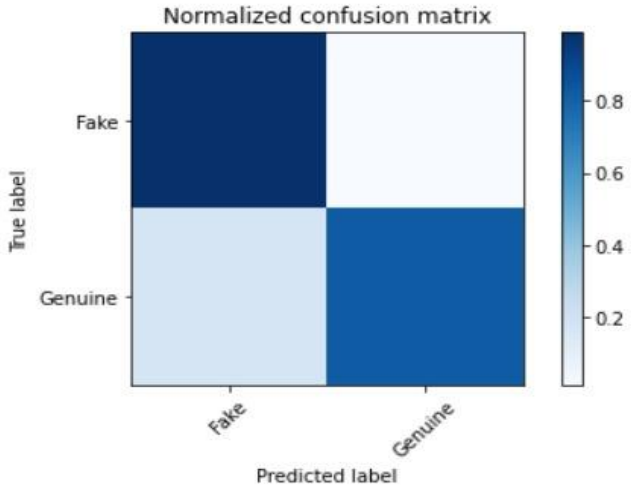
	Predicted FAKE	Predicted GENUINE
Actual FAKE	265	3
Actual GENUINE	53	243

Test Results as depicted by the Confusion Matrix

1. 265occurrences when the model predicted a Fake account as Fake
2. 243occurrences when the model predicted a Genuine account as Genuine
3. 3 occurrences when the model predicted a Fake account as Genuine
4. 53 occurrences when the model predicted a Genuine account as Fake

2. Normalised Confusion Matrix

```
Normalized confusion matrix
[[0.98880597 0.01119403]
 [0.17905405 0.82094595]]
```



	Precision	recall	f1-score	support
Fake	0.83	0.99	0.9	268
Genuine	0.99	0.82	0.9	296
Accuracy			0.9	564
Macro avg	0.91	0.9	0.9	564
Weighted avg	0.91	0.9	0.9	564

Classification Report

The macro-averaged F1 score (or macro F1 score) is computed using the arithmetic mean (unweighted mean) of all the per-class F1 scores. This method treats all classes equally regardless of their support values.

3. ROC (Receiver Operating Characteristic)

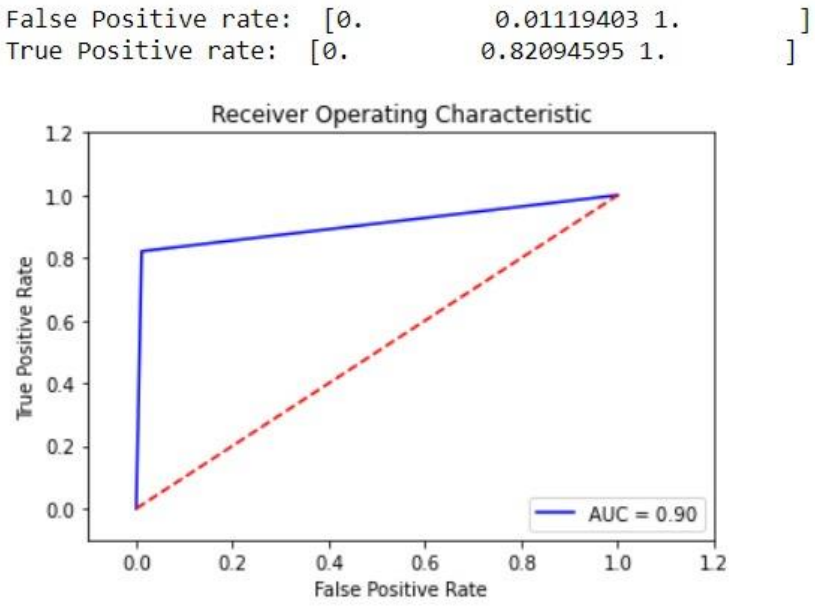


Figure 5.4: ROC curve

We plot the ROC (Receiver Operating Characteristic) curve to show the performance of a classification model at all thresholds. It plots two parameters:

3. True Positive Rate (0.01119403)
4. False Positive Rate (0.82094594)

The efficiency of the Support Vector Machine Classifier in classifying data is **95%**. We have taken 80% of the data for the training dataset and 20% for the testing dataset.

Libraries and Dependencies

1. numpy
2. pandas
3. matplotlib
4. gender_guesser
5. Scikit learn
    - randomForestClassifier
    - svm

    Results

    - Roc curve
    - confusion_matrix
    - auc

Conclusion

We have given a framework using which we can identify fake profiles in any online social network by using Random Forest Classifier with a very high efficiency as high as around 95%. This method can be extended on any platform that needs binary classification to be deployed on public profiles for various purposes.

The model presented in this project demonstrates that Support Vector Machine (SVM) is an elegant and robust method for binary classification in a large dataset. Regardless of the non-linearity of the decision boundary, SVM is able to classify between fake and genuine profiles with a reasonable degree of accuracy (>90%).

Future Work

Fake profile Identification can be improved by applying NLP techniques and Neural Networks to process the posts and the profiles. In the future, we wish to classify profiles by taking profile pictures as one of the features

Since we have limited data to train the classifier, our approach is facing a high variance problem which can be observed in the learning curve as follows High variance problems can usually be mitigated by increasing the size of the dataset which should not be of much concern to Social Networks Organizations which already have fairly large datasets

Bibliography

- https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm
- https://en.wikipedia.org/wiki/Random_forest#:~:text=Random%20forests%20or%20random%20decision,class%20selected%20by%20most%20trees
- https://en.wikipedia.org/wiki/Confusion_matrix

Plagiarism Report

We hereby, declare that the material/ content presented in the report are free from plagiarism and is properly cited and written in my own words. In case, plagiarism is detected at any stage, I shall be solely responsible for it. A copy of the Plagiarism Report is also enclosed.

Report Title:	ML cyber security report
Report Link: (Use this link to send report to anyone)	https://www.check-plagiarism.com/plag-report/869600f727e79fb374062d61762f9d27eb6631659179852
Report Generated Date:	30 July, 2022
Total Words:	209
Total Characters:	1269
Keywords/Total Words Ratio:	0%
Excluded URL:	No
Unique:	94%
Matched:	6%

Contributors

Enrolment No.	Student Name	Roles and Responsibility
00501032020	PRAGATI GANGWAR	Worked on report tables and graphs and report content; worked on SVM algorithm, code debugging.
02801032020	KHUSHI ARYA	Worked on basic theory in report; worked on SVM algorithm, core concept.
02901032020	PARIDHI JAIN	Worked on report formatting and display; worked on code majorly for SVM and code debugging in Random Forest.
03801032020	SNEHA KUMARI	Worked on algorithm theory in report and synopsis; worked on Random Forest code debugging.
03901032020	KHYATI GUPTA	Worked on result and graph explanations in report; worked on code debugging for Random Forest and also with core concept explanation.
04801032020	SOUMMYA PAL	Worked on basic theory in report; worked on Random Forest core concept.