INFORMATION SECURITY MANAGEMENT

Project Review – 2

MACHINE LEARNING TECHNIQUES FOR DETECTING FAKE NEWS

Team Members –

Garvit Doda 19BIT0023

Pranav Mathur 19BIT0040

Priyanshu Raj 19BIT0064

Rohin Srivastava 19BIT0177

Sayyam Maske 19BIT0045

OVERVIEW

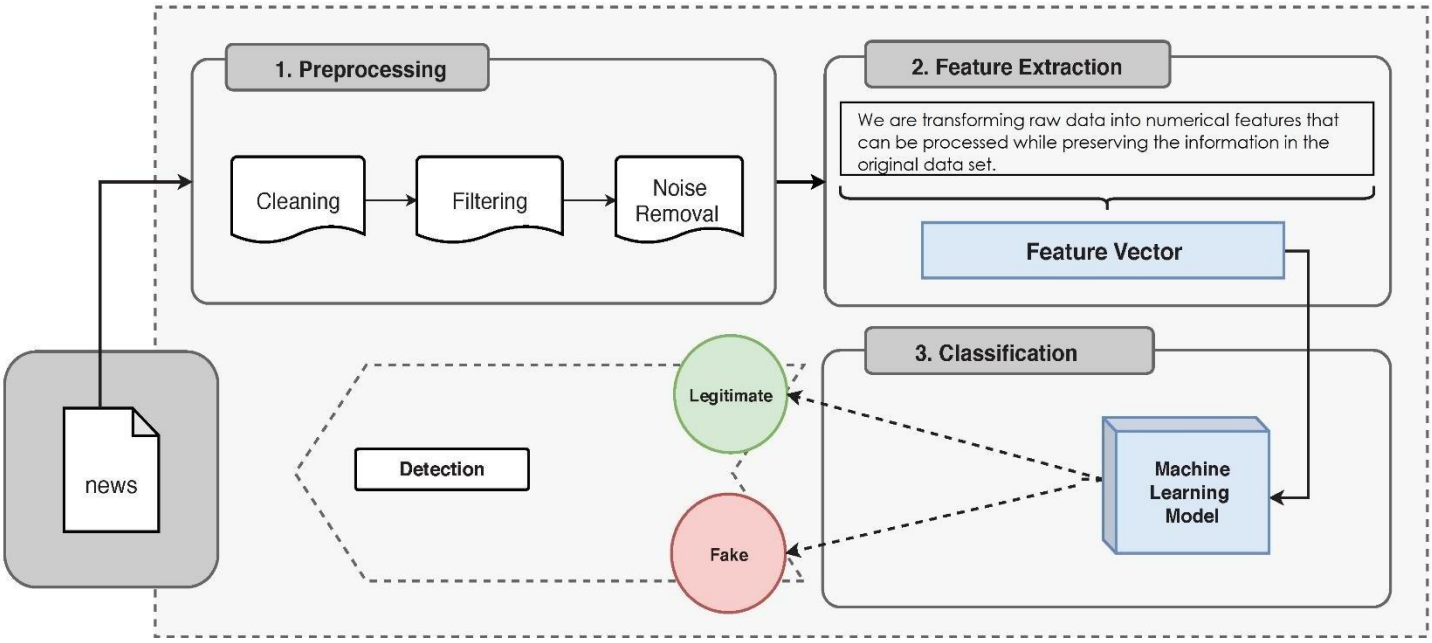
Detection of fake news online is important today as fresh news content is rapidly being produced because of the abundance of available technology. With the advancement of technology, digital news is more widely exposed to users globally and contributes to the increment of spreading hoaxes and disinformation online. Fake news can be found through popular platforms such as social media and the Internet. This paper proposes an improved stacking outfit strategy for fake news detection. A classifier approach was used for improving fake news detection was presented and explained. Use of SVM, Random Forest and Logistic Regression, Decision Tree, Naïve bayes is prominent. We compare the accuracy of these algorithms in this review.

APPROACH

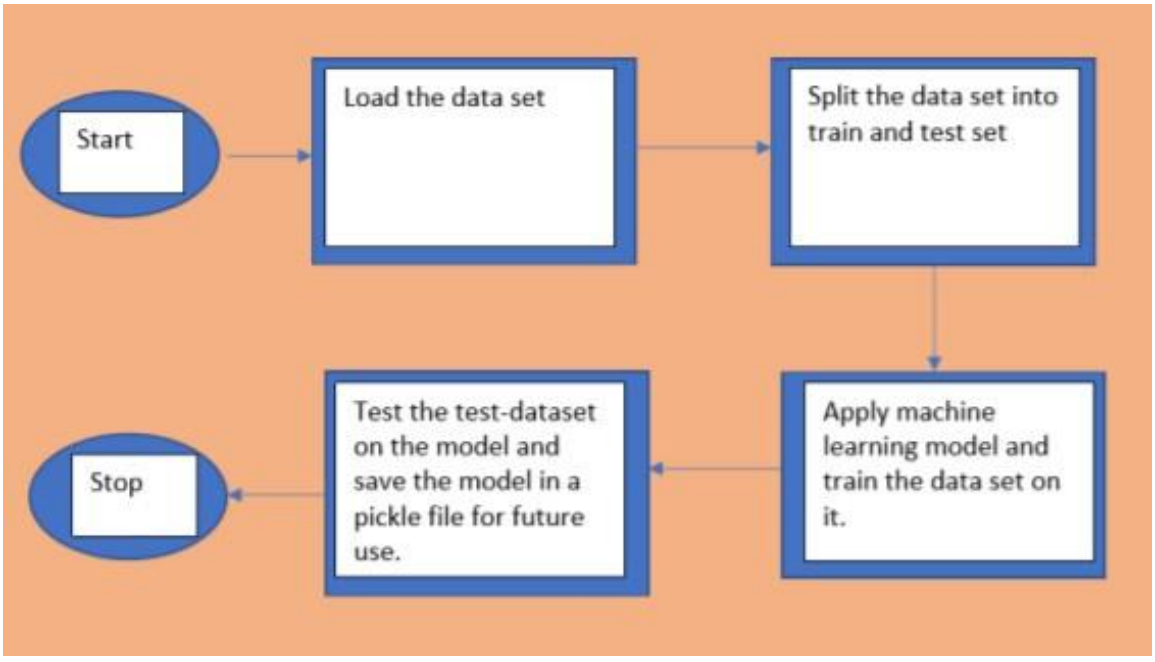
Below mentioned are the steps involved in the completion of this project:

- Collect dataset containing fake and real news from the open-source platforms.
- Write a code to extract the required features from the database.
- Divide the dataset into training and testing sets.
- Run selected machine learning and deep neural network algorithms like SVM, Random Forest, etc. on the dataset.
- Write a code for displaying the evaluation result considering accuracy metrics.
- Compare the obtained results for trained models and specify which is better.

HIGH LEVEL DESIGN



LOW LEVEL DESIGN



ALGORITHM AND METHODOLOGY

Random Forest

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems.

Random Forest Implementation to Test Accuracy and Other Parameters –

```
(base) C:\Users\sriro>python C:\Users\sriro\Desktop\ISMP\RandomForest.py
Accuracy: 99.27%
Confusion Matrix
 [[4674   40]
 [  26 4240]]
Runtime of the program is 88.77261924743652
Matthew Correlation Coefficient is 0.9852711015941262
```

Support Vector Machine (SVM) –

"Support Vector Machine" (SVM) is a supervised machine learning algorithm that can be used for both classification and regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is several features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well. Types:

1. Linear SVM- Hard-margin & Soft-margin
2. Non-Linear

SVMs can be used to solve various real-world problems:

- SVMs are helpful in text and hypertext categorization, as their application can significantly reduce the need for labeled training instances in both the standard inductive and transudative settings. Some methods for shallow semantic parsing are based on support vector machines.
- Classification of images can also be performed using SVMs. Experimental results show that SVMs achieve significantly higher search accuracy than traditional query refinement schemes after just three to four rounds of relevance feedback. This is also true for image segmentation systems, including those using a modified version SVM that uses the privileged approach as suggested by Vapnik.
- Classification of satellite data like SAR data using supervised SVM.

SVM Implementation to Test Accuracy –

```
(base) C:\Users\sriro>python C:\Users\sriro\Desktop\ISMP\SVM.py
accuracy: 98.73%
Confusion Matrix
 [[4574   72]
 [  42 4292]]
Runtime of the program is 798.6994462013245
Matthew Correlation Coefficient is 0.974607263186541
```

Logistics Regression –

Logistic Regression is a "Supervised machine learning" algorithm that can be used to model the probability of a certain class or event. It is used when the data is linearly separable, and the outcome is binary or dichotomous in nature. That means Logistic regression is usually used for Binary classification problems. Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a

categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1. The sigmoid function is a mathematical function used to map the predicted values to probabilities. The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form. The S-form curve is called the Sigmoid function or the logistic function.

$$log\left[\frac{y}{1-y}\right] = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

The above equation is the final equation for Logistic Regression.

Logistics Regression Implementation to Test Accuracy –

```
(base) C:\Users\sriro>python C:\Users\sriro\Desktop\ISMP\LogisticsRegression.py
accuracy: 98.83%
Confusion Matrix
 [[4663   58]
 [  47 4212]]
Runtime of the program is 49.478010416030884
Matthew Correlation Coefficient is 0.9765585456660973
```

Decision Tree –

Decision Tree is a **Supervised learning technique** that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where **internal nodes represent the features of a dataset, branches**

represent the decision rules, and **each leaf node represents the outcome.** In a Decision tree, there are two nodes, which arethe **Decision Node** and **Leaf Node.** Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions. It is called a decision tree because, like a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure. To build a tree, **CART algorithm** is used, which stands for **Classification and Regression Tree algorithm.**

Decision Tree Implementation to Test Accuracy –

```
(base) C:\Users\sriro>python C:\Users\sriro\Desktop\ISMP\DecisionTree.py
Accuracy: 99.68%
Confusion Matrix
 [[4651   11]
 [  18 4300]]
Runtime of the program is 61.45957922935486
Matthew Correlation Coefficient is 0.9935325349814551
```

Naïve Bayes –

Naïve Bayes algorithm is a supervised learning algorithm, which is based on **Bayes theorem** and used for solving classification problems. It is mainly used in *text classification* that includes a high-dimensional training dataset. Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.

It is a probabilistic classifier, which means it predicts based on the probability of an object. Some popular examples of Naïve Bayes Algorithm are **spam filtration, Sentimental analysis, and classifying articles**.

Naïve Bayes Implementation to Test Accuracy –

```
(base) C:\Users\sriro>python C:\Users\sriro\Desktop\ISMP\NaiveBayes.py
accuracy: 94.68%
Confusion Matrix
 [[4441  318]
 [ 160 4061]]
Runtime of the program is 47.69496560096741
Matthew Correlation Coefficient is 0.8939385854341242
```