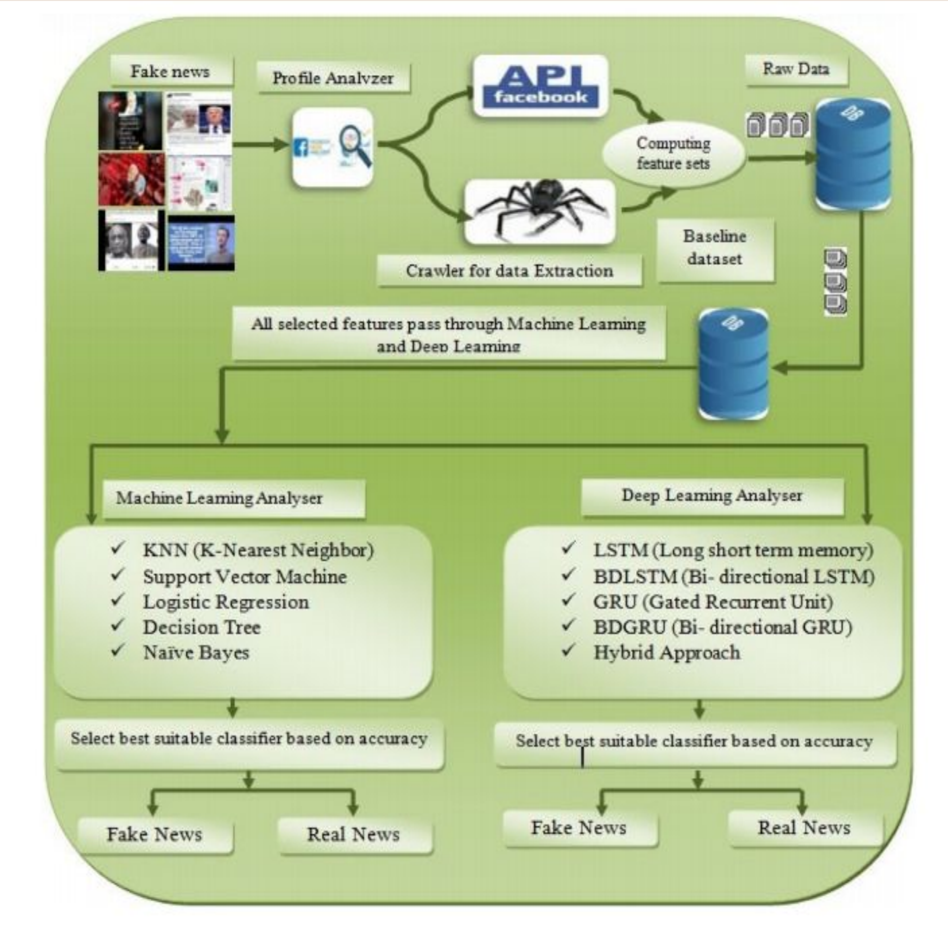
Multiple Features Based Approach for Automatic Fake News Detection

Mohammed Rushad
181CO232

Akshat Nambiar
181CO204

Gist of Paper

- Detecting Fake News based on multiple Features apart from only News Title.
- Obtain Dataset from Facebook by using web crawler built by authors.
- Implement the following models: KNN, SVM, Logistic Regression, Decision Tree, Naive Bayes, LSTM.
- Pick best performing model for Fake News detection.



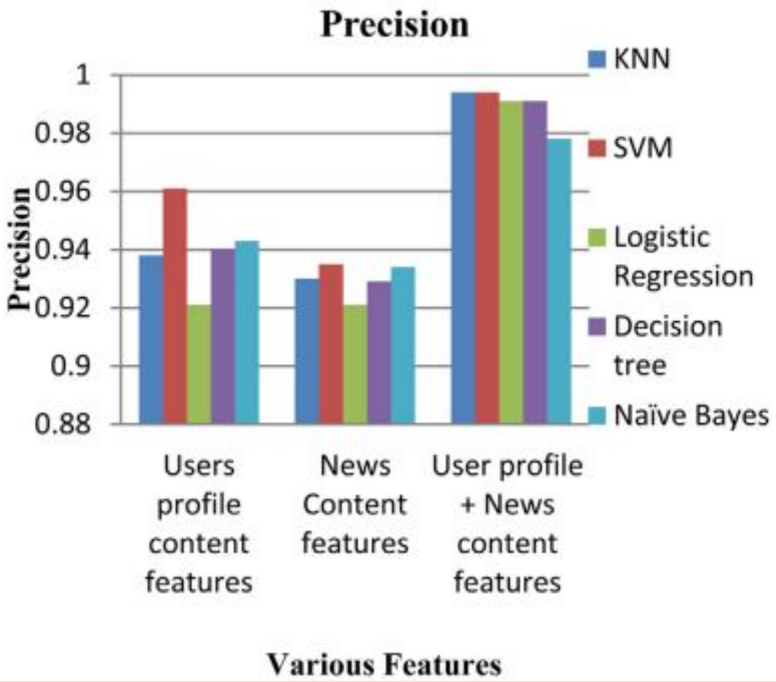
Effectiveness of this method

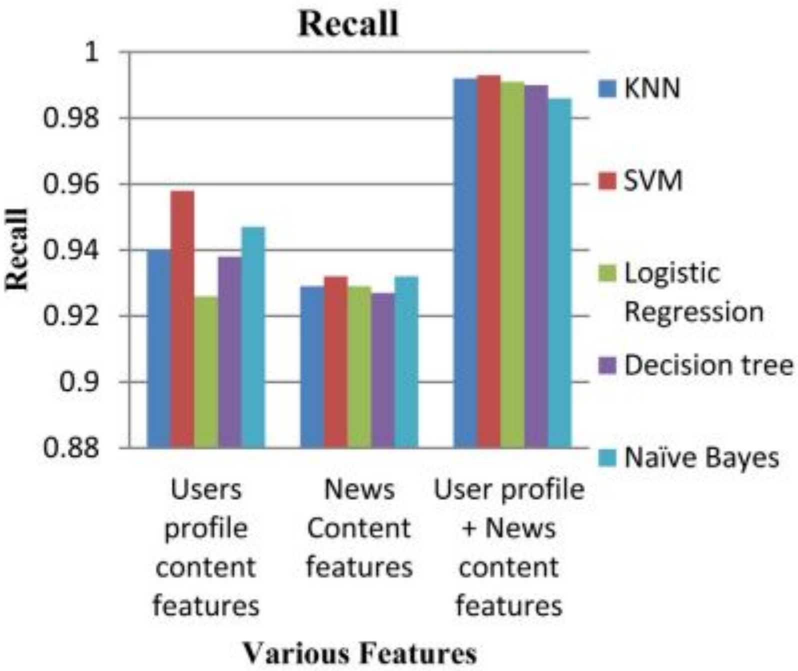
Accuracy of different machine learning and deep learning classifiers.

Features	Measure in %	Machine learning classifiers					Deep learning classifiers
		KNN	SVM	Logistic Regression	Decision tree	Naïve Bayes	LSTM
Users profile content features	Accuracy	94.1	96.2	92.8	94.3	94.8	96.3
News content features	Accuracy	92.3	93.9	92.8	93.0	93.6	91.1
Users profile features + News content features	Accuracy	99.3	99.3	99.0	99.1	98.6	99.4

- User Features method slightly outperforms traditional News Features method.
- Combined, we see a significant increase in performance for all models.

Other metrics to prove success of Method





Paper Dataset

User Features

Feature	Description
Profile ID	Every profile has one unique profile ID.
Profile name	Profile name of the user varies based on users choice
Date of join	It describes how old the profile is.
All friends	Total number of friends of the user.
Profile picture	It shows pictorial identity of the user.
Number of group join	Number of different group's user participated.
Number of page like	This feature shows association of the user in different contents.
News post	It identifies the interest of the user and the multiple events user have participated.
Profile with photo guard	To protect the image from unauthorized user, user uses safety principle as photo guard.
Number of stories shared	User shared the multiple events as story in their profile to view by multiple users.

News Features

Feature	Description
Source	It describes the author or publisher of the news article.
Headline	It describes main highlight of the topic and catch the reader's attention.
Body text	Text that elaborate the detail story of the topic. It also highlights the angle of publisher.
Text	It highlights the story as textual representation i.e. ir readable format
Images (Image with text, image with hyperlink)	Content of the shared news article that provides visual description about that activities or events. It also posted by some user including caption as text and links.

Our Dataset

Features

uuid
ord_in_thread
author
published
title
text
language
crawled
site_url
country
domain_rank
thread_title
spam_score
main_img_url
replies_count
participants_count
likes
comments
shares
type

- Traditional Fake News datasets have mainly 2 Features: Title, Text
- Paper Dataset: Uses 15 features:
  - 10: User Features
  - 5: News Features
- Our chosen dataset uses 11/20 features:
  - 6: Article Performance features
  - 3: News features
  - 2: Misc. features

- Different from Traditional methods.
- Focuses on Article Performances as major indicator.
- This makes the method slightly different from authors' method, since Viewer User Data is given more attention than Poster User Data.

Labels in Chosen Dataset

```
Types of stories type
bias            56
bs            1726
conspiracy     117
fake             3
hate            60
satire          38
dtype: int64
```

The following transformations have been applied on the labels as Fake and Real news

- State -> real
- Satire -> real
- Juncksci -> real
- Hate -> real
- Fake -> fake
- Conspiracy -> fake
- Bs -> fake
- Bias -> real

Project Overview

- Import Dataset
- Replace Labels and Dataset Pre-processing
- NLP Pre-processing
- Creating Final Dataset for Test-Train Split
- Implement Models
- Calculate Accuracy

Preparing the Dataset

1. Clean the Original Dataset.
2. Combining Text and Title features of Original dataset.
3. Performing NLP Processing using CountVectorizer from sklearn to tokenize the words after removing punctuation, stopwords, converting to lowercase etc.
4. Perform Step-3 on all text-based columns of Original Dataset.
5. Combine all Numerical and vectorized text columns into New Dataset.

Models Used

Machine Learning Analysis:

- Logistic Regression

	precision	recall	f1-score	support
0	0.93	0.99	0.96	558
1	0.14	0.02	0.04	42
accuracy			0.92	600
macro avg	0.54	0.51	0.50	600
weighted avg	0.88	0.92	0.89	600

Accuracy for Logistic Regression:
0.9216666666666666

- KNN

	precision	recall	f1-score	support
0	0.93	0.97	0.95	558
1	0.00	0.00	0.00	42
accuracy			0.91	600
macro avg	0.46	0.49	0.48	600
weighted avg	0.86	0.91	0.88	600

Accuracy for KNN:
0.905

Models Used – Continued

- SVM

	precision	recall	f1-score	support
0	0.94	0.96	0.95	558
1	0.25	0.19	0.22	42
accuracy			0.90	600
macro avg	0.60	0.57	0.58	600
weighted avg	0.89	0.90	0.90	600

```
Accuracy for SVM:
0.9033333333333333
```

- Decision Tree

	precision	recall	f1-score	support
0	1.00	1.00	1.00	558
1	1.00	1.00	1.00	42
accuracy			1.00	600
macro avg	1.00	1.00	1.00	600
weighted avg	1.00	1.00	1.00	600

```
Accuracy for Decision Tree using gini Index:
1.0
```

Models Used – Continued

- Naive Bayesian

	precision	recall	f1-score	support
0	0.93	0.99	0.96	558
1	0.33	0.05	0.08	42
accuracy			0.93	600
macro avg	0.63	0.52	0.52	600
weighted avg	0.89	0.93	0.90	600

Accuracy for Naive Bayes:
0.9266666666666666

Models Used – Continued

Deep Learning Analysis:

- LSTM - Long Short-term Memory

Accuracy: 93.00%

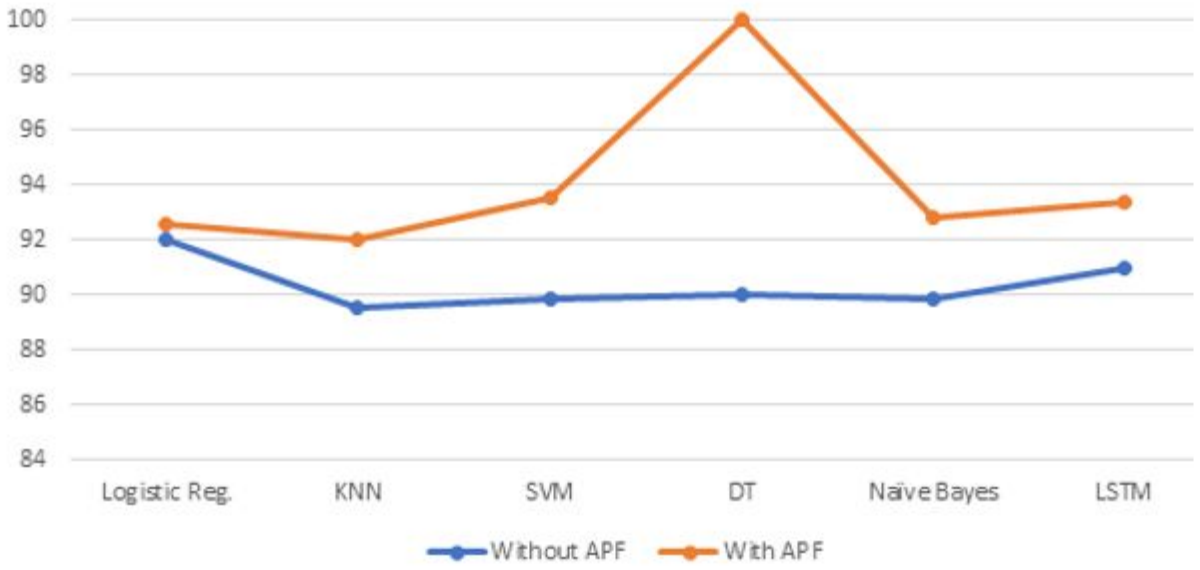
```
Model: "sequential_7"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_7 (Embedding)      (None, 500, 32)           160000
_________________________________________________________________
lstm_7 (LSTM)                (None, 100)               53200
_________________________________________________________________
dense_7 (Dense)              (None, 1)                 101
=================================================================
Total params: 213,301
Trainable params: 213,301
Non-trainable params: 0
_________________________________________________________________
None
Epoch 1/5
22/22 [==============================] - 15s 596ms/step - loss: 0.5544 - accuracy: 0.7687
Epoch 2/5
22/22 [==============================] - 13s 597ms/step - loss: 0.3005 - accuracy: 0.9124
Epoch 3/5
22/22 [==============================] - 13s 581ms/step - loss: 0.2872 - accuracy: 0.9177
Epoch 4/5
22/22 [==============================] - 13s 572ms/step - loss: 0.2506 - accuracy: 0.9321
Epoch 5/5
22/22 [==============================] - 13s 592ms/step - loss: 0.2723 - accuracy: 0.9235
```

Performance Comparison

Algorithm		Accuracy	
		Without AFP (%)	With AFP (%)
Machine Learning			
1	Logistic Regression	92.00	92.66
2	KNN	89.50	92.00
3	SVM	89.83	93.50
4	Decision Tree (gini Index)	90.00	100.00
5	Naïve Bayes	89.83	92.83
Deep Learning			
1	LSTM	91.00	93.33

Performance Comparison Continued

- Therefore, our method shows a consistent increase in accuracy in all models implemented.



Our Contributions

- Traditional Method: News features

  Paper Method: User features + News features

  Our method: Article Performance features + News features + Misc. features

- Method can be altered to detect type of fake news.

Future Work

- Incorporate User + News + Article Performance features
- Incorporate Data from different social networks
- Incorporate model into web-extensions for easier article tagging.

References

Paper: Multiple Features Based Approach for Automatic Fake News Detection (Link)