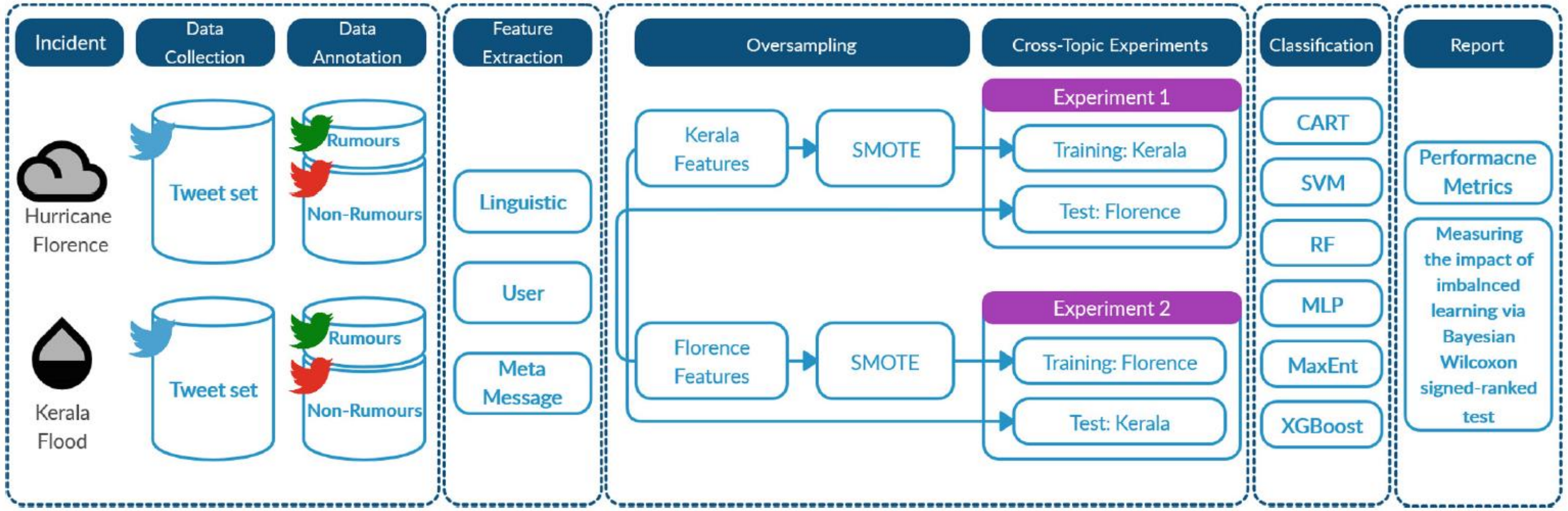
Detecting Rumours in Disasters: An Imbalanced Learning Approach

Amir E Fard
Majid Mohammadi
Bartel van de Walle

TU Delft

Overview

- Motivations
- Data
- Features
- Algorithm
- Feature Selection
- Experiments



Motivations

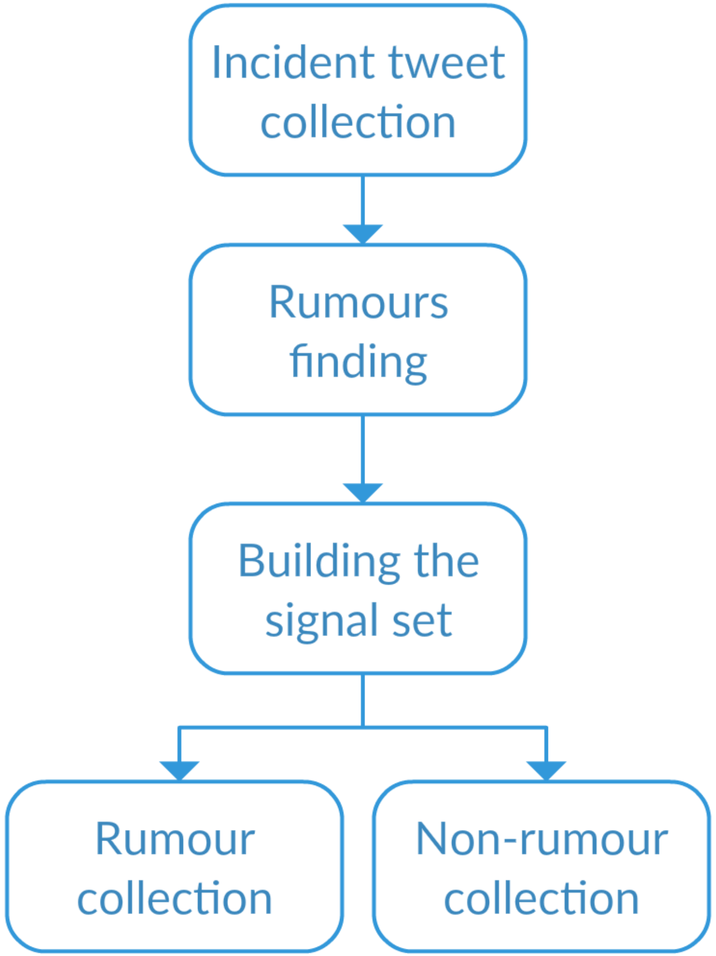
- There is a latent assumption in rumour detection literature
  - The context does not matter and rumour detection over different contexts is the same
- There are a few manuscripts on rumour detection in particular context
  - The context is usually captured by only data.
- In this study, we capture context based on three main components of machine learning:
  - Data + Feature + Algorithm

Data

- For this study, we prepared two datasets regarding the Kerala flood and Hurricane Florence.
- The 2018 Kerala flood was the worst monsoon flooding in a century in Southern India with 400 fatalities and $2.7bn worth of damages.
- The Hurricane Florence was a category four hurricane hit Carolinas in the south-east of the United States. The hurricane caused more than 50 fatalities and up to $22bn damages.

Data

- Incident tweet collection: Setting up a streaming API of Twitter
  - We could collect 100,880 tweets regarding Kerala flood and 101,844 tweets for Hurricane florence.
- Rumours finding: Searching several credible news outlets and fact-checking websites (e.g. Snopes, The Washington Post, The Hindu, and Indian Express).
  - Three rumours in Kerala flood + four rumours in Hurricane Florence with a high level of consistency among different news outlets and fact-checking sources.
- Building the signal set: Extraction of the rumour-related tweets corresponding to these events if the tweet contains the keyword relevant to the rumour. The tweets without explicit keywords would carry non-rumour label.
  - Kerala dataset => 2,000 rumour related and 98,880 non-rumour-related tweets.
  - Florence dataset => 2,382 rumours and 99,462 non-rumour tweets.



Features

- 83 features (linguistic, user, and meta)
- They are related to disaster and either taken from the literature of computational rumour detection or introduced in this work Early-available features
- No changing feature

Number of exclamation marks in a tweet
Number of question marks in a tweet
Number of characters in a tweet
Number of words in a tweet
Number of uppercase letters in a tweet
Number of lowercase letters in a tweet
Number of first person pronoun in a tweet
Number of second person pronoun in a tweet
Number of third person pronoun in a tweet
Number of capital words in a tweet
Average word complexity in a tweet
Number of vulgar words in a tweet
Number of abbreviations in a tweet
Number of emojis in a tweet
Polarity of a tweet
Subjectivity of a tweet
Tone of a tweet
Positive words score of a tweet
Negative words score of a tweet
† Frequency of Part of Speech (POS) tags in a tweet (19 features)
◇ Frequency of Name Entity Recognition (NER) tags in a tweet (17 features)
Opinion and insight score
Anxiety score
Tentativeness score
◇ Certainty score
Sentence complexity

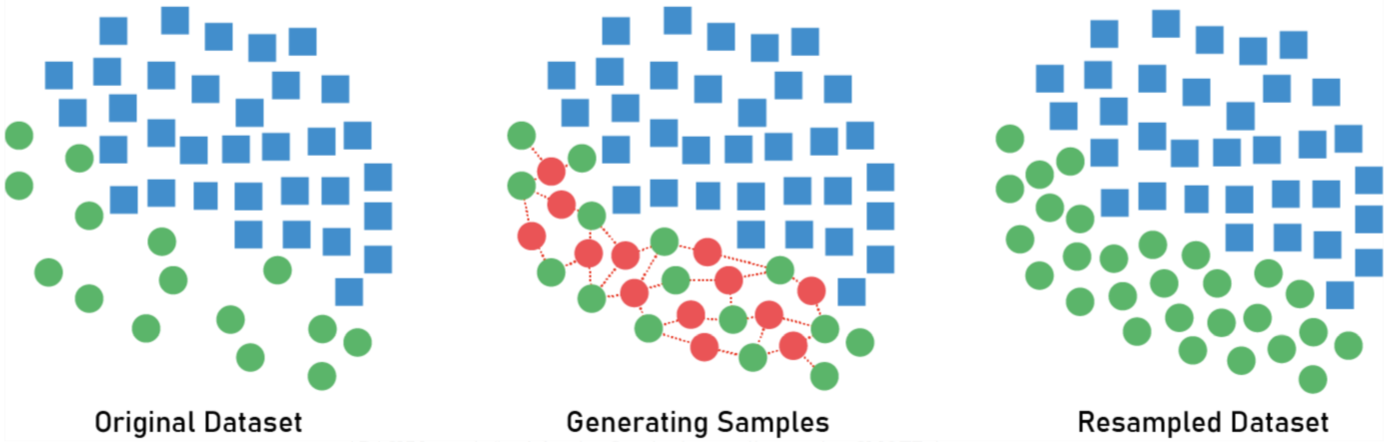
Profile description (binary)
Verified account (binary)
Number of Statuses
Influence
Number of following
User role
◇ Attention
Account age (day)
◇ Openness (binary)
Profile location (binary)
† Profile picture (binary)
Profile URL (binary)
◇ Average follow speed
◇ Average being followed speed
◇ Average like speed
◇ Average tweet speed
† Screen name length
† Number of digits in screen name

Number of hashtags in a tweet
Number of mentions in a tweet
Tweet URL (Binary)
Number of multimedia in a tweet
◇ Location sharing (binary)

TUDelft

Algorithm

- Due to the imbalance nature of the annotated dataset, we use imbalance learing.
- There are different approaches for imabalnce learning. We choose to use oversampling using synthetic minority oversampling technique (SMOTE) technique.

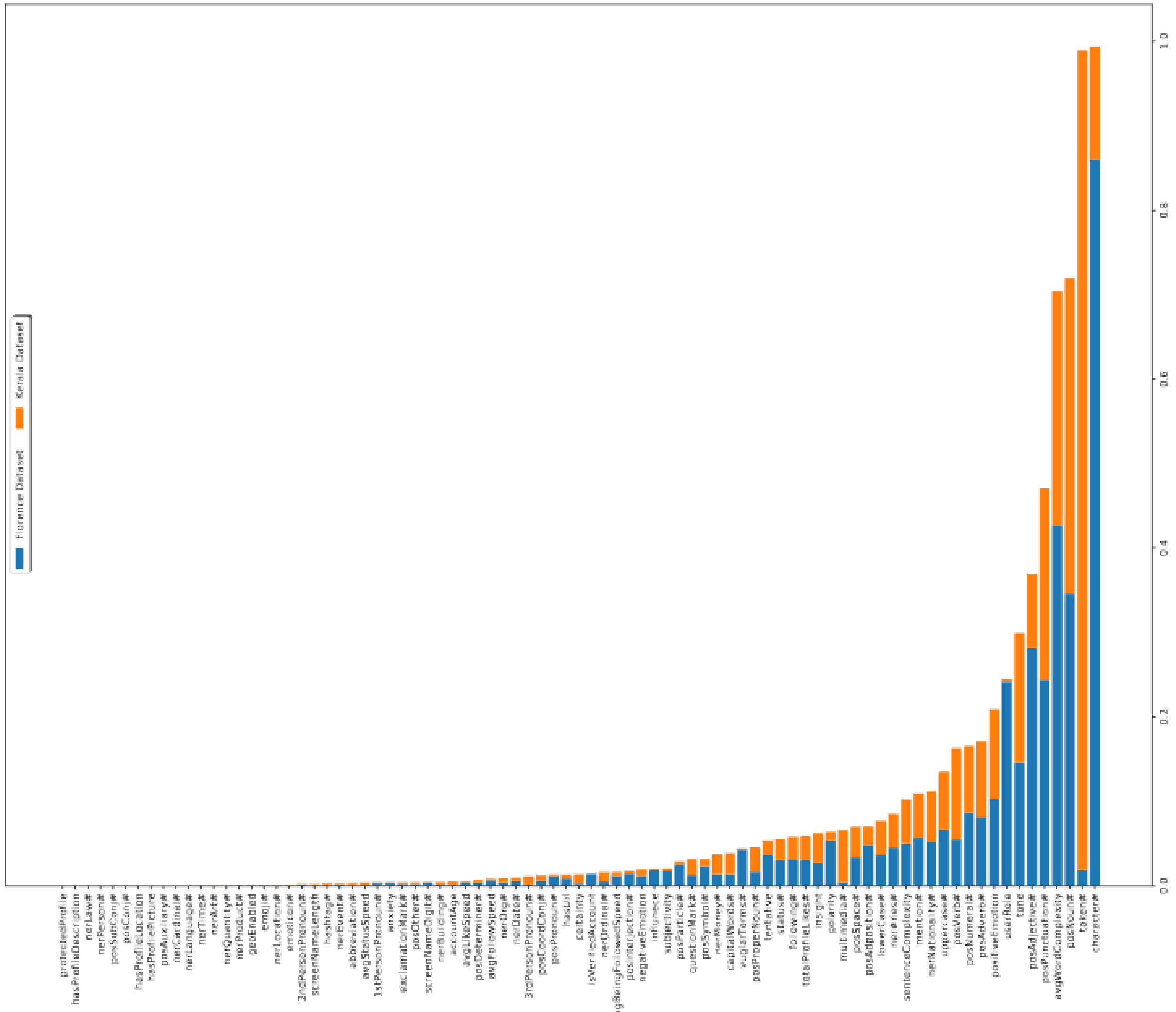


ADASYN stands for Adaptive Synthetic sampling, and as SMOTE does.

Feature selection

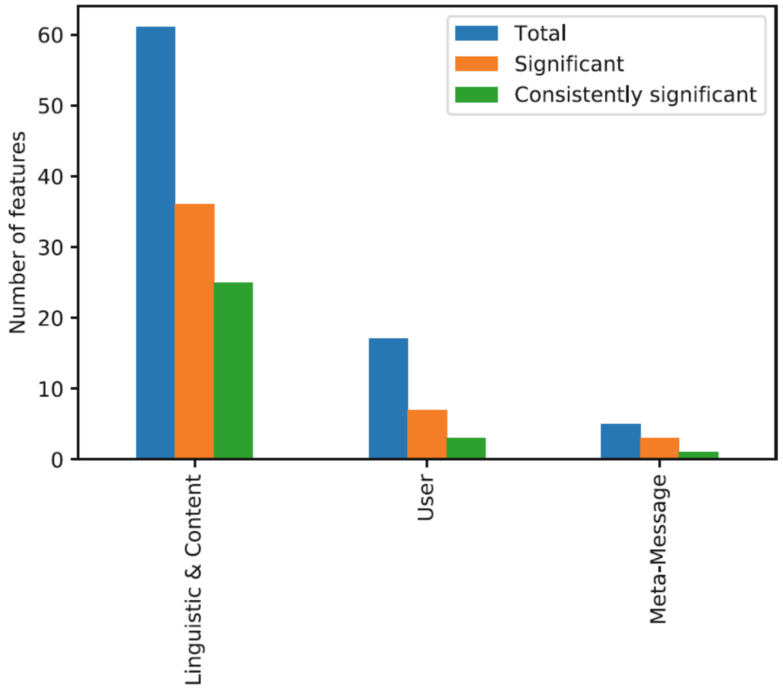
- Feature score => using random forest, XGBoost, adaptive boosting, regression tree, and extremely randomized trees as classification algorithms with an embedded feature selection mechanism.
- By summing up the features weights we obtain the degree of significance for each feature.
- To find the significant features we determine a threshold for the features score (τ=0.001).
  - If a feature_score < τ then feature is insignificant
  - If a feature_score >= τ then feature is significant
    - If a feature_score in both datasets >= τ then feature is consistently significant.

Feature selection



Feature selection

- The following Figure illustrates the number of significant and consistently significant features as well as the total number of features for each feature category.



TU Delft

Experiments

- In this study we use seven classifiers belonging to different learning paradigms:
  - multi-Layer perceptron (MLP) as a neural network,
  - support vector machine (SVM) as a kernel machine,
  - classification and regression trees (CART) as a decision tree,
  - random forest (RF) as an ensemble of trees,
  - XGBoost as a boosting approach, and
  - maximum entropy (MaxEnt) as an exponential model.

- We use oversampling on the training set (It is only for training data, in other words, the test dataset is still intact and preserves its imbalanced shape.)

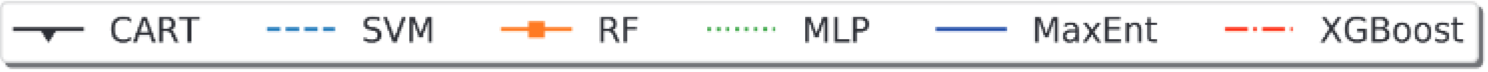
- We use one of our datasets for training and the other one for testing; then we switch the training and test set to assess the robustness of the proposed approach

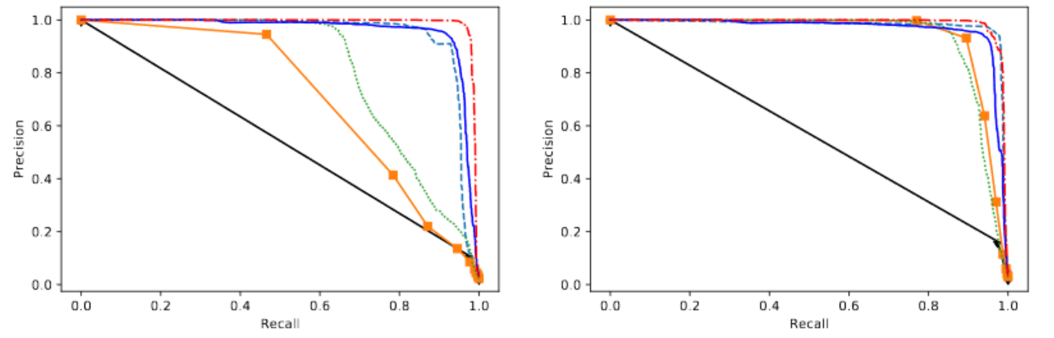
Experiments

Classifiers	PR	RE	F1-Score
CART	90.5%	91.7%	91.1%
SVM	92.6%	55.5%	69.4%
RF	93.9%	92.4%	93.1%
MLP	92.6%	90.6%	91.6%
MaxEnt	91.2%	79%	84.7%
XGBoost	95%	94.4%	94.7%
CRF [34]	66.7%	55.6%	60.7%

TU Delft

Experiments

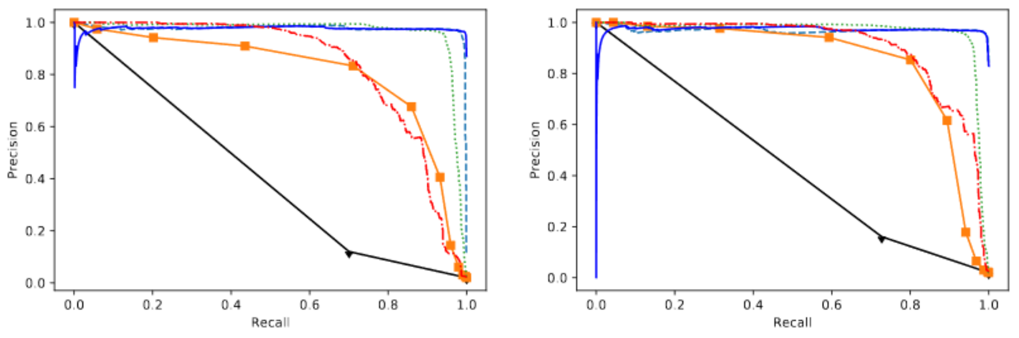




(a) Learning with imbalanced data when the training is with Kerala and test with Florence (Kerala ⇒ Florence)

(b) Learning with balanced data when the training is with Kerala and test with Florence (Kerala ⇒ Florence)

Training: Kerala => Test: Florence



(c) Learning with imbalanced data when the training is with Florence and test with Kerala (Florence ⇒ Kerala)

(d) Learning with balanced data when the training is with Kerala and test with Florence (Kerala ⇒ Florence)

Training: Florence => Test: Kerala

Experiments

	Classifiers	Training with imbalanced data (AUPRC)	Training with balanced data (AUPRC)
Kerala ⇒ Florence	CART	54.2%	56.8%
	SVM	94.6%	97.8%
	RF	71.5%	94.3%
	MLP	80.9%	93.3%
	MaxEnt	95.9%	96.5%
	XGBoost	98.7%	98.8%
Florence ⇒ Kerala	CART	41.4%	44.6%
	SVM	97.7%	97%
	RF	81.3%	85.5%
	MLP	96.3%	96.4%
	MaxEnt	97.6%	97.2%
	XGBoost	83.9%	89.2%

Future directions

In order to have a universal system for rumour detection, we need to know the behaviour of rumours in different subject domains. Performing domain specific rumour analysis on various subject domains would be a practical step toward discovering the behaviour of rumours.

TU Delft

Thank you!

a.ebrahimifard@tudelft.nl
@amiref_1991