Early Detection of Social Media Hoaxes at Scale

ARKAITZ ZUBIAGA, Queen Mary University of London, United Kingdom
AIQI JIANG, Queen Mary University of London, United Kingdom

The unmoderated nature of social media enables the diffusion of hoaxes, which in turn jeopardises the credibility of information gathered from social media platforms. Existing research on automated detection of hoaxes has the limitation of using relatively small datasets, owing to the difficulty of getting labelled data. This in turn has limited research exploring early detection of hoaxes as well as exploring other factors such as the effect of the size of the training data or the use of sliding windows. To mitigate this problem, we introduce a semi-automated method that leverages the Wikidata knowledge base to build large-scale datasets for veracity classification, focusing on celebrity death reports. This enables us to create a dataset with 4,007 reports including over 13 million tweets, 15% of which are fake. Experiments using class-specific representations of word embeddings show that we can achieve F1 scores nearing 72% within 10 minutes of the first tweet being posted when we expand the size of the training data following our semi-automated means. Our dataset represents a realistic scenario with a real distribution of true, commemorative and false stories, which we release for further use as a benchmark in future research.

1 INTRODUCTION

Social media platforms such as Twitter, Instagram and Facebook are increasingly being used by the general public to follow the latest news [25, 45] and by journalists for newsgathering [12, 14, 63, 66]. The fact that anyone can post and share content in social media without moderation enables decentralised production of citizen journalism with an unprecedented detail of report [6]. However, the unmoderated nature of social media also leads to the production and diffusion of hoaxes [2, 31], which exacerbates the credibility of social media as a source for news consumption. With as many as 62% of citizens using social media for news consumption in 2016 in the US [18], verification is becoming increasingly important to avoid the spread of misinformation [20]. This problem is producing an increasing interest in the scientific community to develop automated systems that can determine the accuracy of social media reports with the aim of getting rid of hoaxes [1, 59, 64].

Research in automated detection of misinformation in social media has indeed increased in recent years [48, 60, 64]. Researchers have assessed the capacity of average people to identify reports that are inaccurate, finding that their performance leaves much to be desired [24]. This reinforces the need to develop automated systems for disinformation detection, however existing work has largely limited to post-hoc classification of reports as true or false, i.e. aggregating an entire timeline of tweets. This means that reports can only be classified hours or even days after they are first released. Research in performing early classification of reports by their truth value is very scarce, partly because of the limited availability of large-scale datasets for the task. An important challenge that hinders the development of early hoax detection systems is the dearth of suitable datasets. Datasets are usually produced by first identifying lists of fake reports. These are then completed by including news reports from other sources to have balanced datasets with fake and real news reports. This, however, is not necessarily representative of a real scenario of incoming reports. This work aims to overcome this issue by introducing a novel approach for generating a large-scale, representative dataset with accurate reports and hoaxes.

Authors' addresses: Arkaitz Zubiaga, a.zubiaga@qmul.ac.uk, Queen Mary University of London, London, United Kingdom; Aiqi Jiang, a.jiang@qmul.ac.uk, Queen Mary University of London, London, United Kingdom.

In this work we focus on the diffusion of hoaxes and accurate reports of the same type, in an effort to build a representative dataset of accurate and inaccurate reports. A hoax can be defined as a fabricated story that intends to deceive others, such as reporting the existence of a bomb to the police when there isn't one. The word 'hoax' originates from the Latin verb *hocus*, meaning "to cheat" [37]. In the scientific literature, a 'hoax' has been defined by MacDougall as 'a deliberately concocted untruth made to masquerade as truth' [30].

To develop a representative data collection process collecting hoaxes and a set of comparable truthful reports, we look into death reports of celebrities circulating in social media. Death reports are known to be riddled with hoaxes,[1] users frequently making up the death of celebrities, making them viral as if they were real reports and ultimately deceving others. We match these death reports in social media with the entry of the person in question in the Wikidata knowledge base [55]. While we conduct experiments online to assess the effectiveness of our models for early detection of hoaxes, our methodology enables performing offline annotations of the data, building a static dataset which we then test simulating a streaming scenario. The advantage of performing the annotation work offline is that we can determine with confidence whether the person really died or not, once the veracity value of the story is settled and its entry in Wikidata is up-to-date. This annotation process has the advantage of being a semi-automated procedure, for instance because one can automatically determine if a person died if the death date on Wikidata matches the date of the tweets reporting the death. When there is no match, it requires more careful manual analysis to determine if it is a hoax, and hence the semi-automated nature of the task, rather than full automation. This semi-automated dataset generation process enables us to create a large-scale dataset with 4,007 death reports over the course of three years (which have over 13 million tweets associated). This dataset can then be exploited in a streaming scenario to determine the earliness with which our system can make accurate predictions on the veracity of the death reports.

In this paper we make the following key contributions:

- We propose a novel semi-automated method that leverages the Wikidata knowledge base to build a large-scale dataset for early detection of hoaxes in social media.
- We perform experiments using class-specific representations of word embeddings for effective detection of hoaxes. This approach is possible thanks to the semi-automated approach for generation of large-scale datasets, which enables large sets of training data to be available for training the word embedding models of our classifier.
- We broaden our set of experiments by looking into the impact of the size of the training data on the classifier's performance.
- We look into the use of sliding windows which enables us to leverage the most recent tweets in the timeline associated with a report, instead of the entire timeline. This is motivated by the hypothesis that social media users may exhibit a self-correcting behaviour in these situations, where users may change their mind over time as new reports come out.
- We perform an analysis of the social features used in our experiments, which provides insights into the diffusion of death reports, with a focus on distinguishing between hoaxes and accurate reports.

Our experimentation shows the effectiveness of our proposed approach for building class-specific word representations, achieving F1 scores of 72% within just 10 minutes of the first report being posted, and outperforming other baselines. Our experiments also show that the use of sliding windows does not help improve the results; instead, the entire stream of tweets available at the time of classification leads to substantially better results than restricting it to sliding windows of the most recent tweets, hence not validating our hypothesis of a self-correcting behaviour. We

[1] http://www.snopes.com/tag/celebrity-death-hoaxes/

also observe that it is important to have a reasonably sized training set to achieve competitive results, with results beginning to plateau only when more than 21 months' worth of data is used for training.

The release of our dataset and trained word embedding models further enable research in veracity classification using a benchmark scenario.

2 RELATED WORK

2.1 Veracity Classification

Research in determining the accuracy of reports in social media has focused on two different directions: classifying the perceived credibility level of information in social media [7, 19, 35, 46, 57], and classifying the veracity of social media reports into one of true or false [42, 48, 52, 62, 64]. While the objective in the former is to try to determine the subjective perception of the veracity of a report by its recipients, our objective here aligns with that of the latter, i.e. determining the objective veracity of reports. This is very important for social media users, as it can help flag reports that are classified as fake, as well as to validate reports found to be accurate. Assistance with verification of information in social media is key as previous research found that social media users struggle to identify when information is false [24, 65].

Previous work on veracity classification has used different social media platforms including Twitter [50] and Sina Weibo [58]. However, most of this work has performed post-hoc classification of reports as true or false [21, 29, 49], which means that they need to observe the entire development of a story before classifying it. This may imply hours or even days of delay by the time a story can be classified. Our objective here instead is to aim for early classification of stories, with the ultimate goal of detecting hoaxes early on.

Research looking into either real-time or early detection of hoaxes is scarce. [27] use a set of features including user metadata and propagation structure to verify stories within hours of being posted for the first time. They show competitive performance with the use of both feature sets 72 hours after the story was first posted. Another approach is presented by [44], combining hashtags and links as features to determine the veracity of reports. They report results between 1 and 10 hours, with results increasingly improving over time. While both of these are clever approaches that are worthwhile considering, neither of their systems was publicly released and the features used in their experiments are not reproducible with the level of detail provided.

There is also work tackling early detection of "fake news." [28] define the early detection task as that consisting in determining the truth value of a single tweet. They look at the propagation of the tweet through retweets, using the user profiles to try to determine early on the veracity value of the tweet. This method is limited to retweets of a single tweet, hence the authors only look at features from the profiles of those retweeting; this differs from our study where we consider multiple tweets associated with each story (i.e. a death report), and therefore our experiments look at the capacity of determining the veracity of a story by aggregating related tweets over time in a streaming scenario. In [61], the authors used a dataset of tweets with associated fact-checks from professional organisations. This is also a sensible methodology, to which our methodology contributes by defining a novel methodology to come up with a large-scale dataset with annotations grounded in associated Wikipedia pages, for a type of story in which the veracity can easily be determined with confidence post-hoc, celebrity death reports. In other cases, researchers have referred to their task as "early detection of fake news," as is the case with Gereme et al. [15], it is however unclear how the temporal aspect of the stories has been incorporated into their models, as it is never discussed.

Others have taken a different approach by using stance classifiers [11, 13, 17, 40, 67, 68]. Instead of using a classifier that directly outputs one of true or false given a report as input, they try to determine the stance that each social media post expresses with respect to a report, such as supporting, denying, querying or commenting. They then propose to aggregate the different stances to determine the likely veracity of a report. While this is a sensible approach, it also requires a large amount of posts to be observed in order to aggregate the different stances, which may impede early determination of report veracity.

Research in early detection of veracity in social media is still limited, largely hindered by the lack of suitable datasets that enable experimentation in a streaming scenario. Our benchmark dataset and experimentation aims to fill this gap.

2.2 Related Datasets

Research in veracity classification has been largely limited by the dearth of proper datasets. This is changing in recent years, however often with limitations in representativity of the dataset contents or quality of veracity annotations. As [48] stated, development of a dataset annotated for veracity is very challenging, as judgments from professionals are generally needed to carefully verify and subsequently annotated stories. As shown by previous research [24], average users struggle to distinguish true and false stories. It is therefore not generally a suitable task to be performed through crowdsourcing, requiring careful analysis of stories either through professional input or by checking reputable sources or evidence. As a result, few representative datasets have been produced. Most of these datasets are created by first collecting false stories, and then completing the datasets with randomly picked true stories [26, 27]. The use of different methodologies for collecting false and true stories is however not ideal as it will inevitably differ from a real scenario. Furthermore, existing datasets are normally made of isolated posts annotated for veracity (cf. [49, 56]), which pose limitations when one wants to investigate the earliness of veracity classification models in a incoming stream of multiple posts linked to a single story. To test our models at different points in time on a streaming scenario, we need to collect a timeline of tweets linked to each story instead.

Recent years have seen a surge of datasets to research in the misinformation landscape, dominated mostly by those containing isolated posts and hence not enabling research in early detection. The vast majority of these datasets are made of fact-checks collected from professional organisations, with claims or headlines labelled as true or false (or a wider spectrum with combinations of these, such as mostly true, mostly false and half true). This is the case of NELA-GT-2018 [38] and FakeNewsNet [47] in English, GermanFakeNC [54] in German and Factck.br [34] in Portuguese. These can be deemed high quality annotations, particularly when they are collected from fact-checking organisations recognised by the International Fact-Checking Network (IFCN),[2] however their representativity can be questionable as it is dependent on the editorial selection of stories by the fact-checking organisation in question. Another dataset for fact-checking claims is FEVER [51], which collected and altered claims extracted from Wikipedia, automatically creating correct and incorrect claims for fact-checking; this is a clever approach to create a large-scale dataset, which however does not enable exploration of earliness in classification due to claims being isolated. Others have relied on the quality of sites to determine if their news articles are real or fake, i.e. a collection of articles from a reputable news organisation (e.g. Wall Street Journal) would be deemed accurate, whereas articles from low quality or parody news outlets (e.g. The Onion) would be deemed fake (cf. FakevsSatire [16], FA-KES [43], Newsbag [22]); this enables easy collection of large-scale datasets, however it raises concerns about the quality of the annotation, as well as whether the final classification task consists in determining the veracity of articles or instead in classifying the source of the news.

[2] https://www.poynter.org/ifcn/

r/Fakeddit [36] provides a large-scale, representative collection of Reddit posts, where labels are however automatically determined by using machine learning models, which cannot guarantee high quality of labels. Credbank [33] is another related dataset, which however includes annotations for perceived credibility scores, rather than actual veracity scores.

Work on the PHEME project [69] focused instead on rumours, i.e. stories that start off as unverified. Through the organisation of two shared tasks, RumourEval [11, 17], the project looked at how the stances expressed by users over time can help determine the veracity of rumours early on. This is one of the most related datasets to the present work, which however does not scale as easily as it required manual input from journalists to determine the veracity of rumours. The data collection and annotation approach defined in this work is semi-automated, enabling generation of large-scale datasets.

In this work, we describe a novel approach for semi-automated dataset generation, which removes the sampling bias as verification of larger sets of instances is possible through the use of Wikidata as an external source. Likewise, our approach enables collections of both true and false stories by following the same methodology, leading to the first large-scale, representative dataset collected out of social media.

2.3 Learning Class-specific Word Representations

Class-specific word representations have been found to be useful for different classification tasks, as is the case with the use of Brown clusters to build class-specific language models [5]. Brown clusters have been successfully used by researchers for training word representations [53], natural language processing tasks such as dependency parsing [23] or for building class-specific language models [3], among others. As a state-of-the-art approach for semantic word representation, here we make use of word embeddings [32]. We propose to train and leverage class-specific word embeddings to learn the patterns of each class in the training data. The difficulty to achieve this generally lies in the necessity for large-scale annotated datasets that have large numbers of instances for each class. Our semi-automated approach for building large-scale annotated datasets enables to have large collections of data to train class-specific word embeddings.

3 MATERIALS AND METHODS

3.1 Dataset

Our data collection methodology is semi-automated, involving little and easy human input, which enabled us to collect a large-scale dataset. The dataset generation process consists of three steps: (1) data collection, (2) linking to Wikidata, and (3) data annotation.

3.1.1 Data collection. We first perform keyword-based collection of tweets from Twitter. We use 'RIP' as a keyword that is largely associated with death reports. Twitter's results are not case sensitive, so we collect all tweets including the keyword and remove those that are not upper-cased at a later stage. We perform the collection of tweets containing the keyword 'RIP' for a period of three years between January 1, 2012 and December 31, 2014. This longitudinal data collection led to a total of over 94.2 million tweets.

3.1.2 Linking to Wikidata. As we completed the collection of tweets at the end of 2014, we downloaded a dump of Wikidata [55] in January 2015, which is a structured knowledge base that includes, among others, an extensive database of notable people, in part extracted from Wikipedia but also completed by volunteer contributors. The entries of these notable people in the knowledge base include their death date, when the person deceased; a null value as the death date

indicates the person is alive. We used its API to download all entries corresponding to people,[3] leading to a collection of 1,136,543 different people. Each of these entries includes the fields shown in the following example:


```
{"id":"8023",
 "name":"Nelson Mandela",
 "birth":{"date":"1918-07-18","precision":11},
 "death":{"date":"2013-12-05","precision":11},
 "description":"former President of South Africa, anti-apartheid activist",
 "aliases":["Nelson Rolihlahla Mandela","Mandela","Madiba"]}
```


We are interested in most of these features for our research, but especially in the name and aliases, which we use to identify mentions of people in our 'RIP' tweets, and also the death date, which indicates if a person is still alive or has died on a particular date. Note that birth and death dates have a precision value associated, which refers to the granularity of the date. A value of 11 implies the date is accurate at the day level. The standard for contemporary people is for this value to be 11. Year and month-level precision scores are occasionally given for people in earlier centuries. We use the Wikidata knowledge base to look for mentions of contemporary people in our Twitter dataset, and so the lack of precision for ancient people does not have an effect in our case.

Having the collection of 'RIP' tweets and the entries for people on Wikidata, we look within the tweets for mentions of names (and aliases) of people in the Wikidata knowledge base, e.g. tweets containing 'RIP Nelson Mandela'. To do so, as a first step, since the keyword search on Twitter is case insensitive, we removed all occurrences where the keyword 'RIP' was not completely upper-cased. We then looked for tweets where the keyword 'RIP' was followed by one of the person names (or aliases) in Wikidata. We do this for all the tweets and keep the instances in which the name of a person is mentioned at least 50 times in a day. Removing instances with fewer than 50 tweets reduces noise from spam tweets that did not go viral, and makes the manual annotation (which we explain below) more manageable. Note that this process can also identify numerous instances of mentions of the same person, i.e., being reported dead in social media more than once within the time frame of our study between 2012 and 2014. Consecutive days mentioning the same person are considered part of the same death instance, while we only consider a new instance when there is at least one day gap between mentions. This process led to a dataset with 4,007 death reports pertaining to 3,066 different people. The total number of tweets associated with these reports amounts to 13,302,600.

3.1.3 Description of the Hoax Detection Task. The hoax detection task consists in identifying emerging reports that are false. In our experiments, we aim to identify the death reports that have been fabricated, i.e. reporting cases of deaths that have not actually happened. We formally define the death hoax detection task as that in which a supervised classifier has to determine which of the following three categories a new incoming reporting belongs to: $Y = \{real, commemoration, fake\}$. We use three categories as we distinguish cases of *fake* reports, where a death has been fabricated, *real* reports, where a death report has indeed recently happened, and *commemorations*, where a past death is being remembered. In what follows we detail the annotation process we relied on.

3.1.4 Annotation. At this stage we have 4,007 death reports linked to Wikidata pages. To conduct the annotation of these death reports, we developed an annotation tool that visualises the stream of tweets associated with a report, along with a form that enables the annotation. Tables 1, 2 and 3 show the information we provide in the annotation tool, with three examples for real, commemorative and fake death reports.

[3]To identify entries that are about people, we looked for entries with the property "P569", which refers to "date of birth" and is therefore indicative of an entry belonging to a person: https://www.wikidata.org/wiki/Property:P569

Death report on: 12th December, 2014	**Wikidata entries**
	#1: personname (death: 12-12-2014, born: 1940)
RIP personname ...	#2: personname (death: 0, born: 1975)
RIP personname ...	
RIP personname ...	⊗ Real
RIP personname ...	◯ Commemoration
	◯ Fake

Table 1. Example of real death report, where the date of the death report and the death date of a Wikidata entry match. Note there are two Wikidata entries matching the person name in question in this case, where the death date of one of them matches that of the death report.

Death report on: 12th December, 2014	**Wikidata entries**
	personname (death: 12-12-2009, born: 1945)
RIP personname ...	
RIP personname ...	◯ Real
RIP personname ...	⊗ Commemoration
RIP personname ...	◯ Fake

Table 2. Example of a commemorative death report, where the date of the death report and the death date of a Wikidata entry are exactly years apart from each other, hence indicating that Twitter users are remembering the person who died years ago.

Death report on: 12th December, 2014	**Wikidata entries**
	personname (death: 0, born: 1972)
RIP personname ...	
RIP personname ...	◯ Real
RIP personname ...	◯ Commemoration
RIP personname ...	⊗ Fake

Table 3. Example of fake death report, where the matching Wikidata entry has no death date (i.e. death date = 0).

Most importantly, the annotation tool shows the date in which the death report broke on Twitter, along with a list of Wikidata entries of candidates matching the person mentioned in the tweets. An exact match between the date of the death report and the death date of one of the Wikidata entries is then highly indicative of a real death report linked to that candidate. Hence, for a majority of the death reports, a tentative annotation candidate can be done automatically by the tool, in the following cases:

- If the date of the death report and the death date of one of the Wikidata entries match, the annotation tool will automatically mark the death as being real. Note that besides exact date matches, we also automatically mark it as a real death if the date of the report and the death date of a Wikidata entry are only one day apart, due to time zone differences (i.e. tweets being UTC and the person dying elsewhere in the world).
- If the date of the death report and the death date of a Wikidata entry match (or they are one day apart) but on a different year, then we automatically mark it as a commemoration.
- If there is a single Wikidata entry listed as a candidate and that entry has not died (death date = 0), then we mark it as fake.

These automated annotations are then shown to the annotator, who supervises and approves (or changes) the annotation, which is substantially faster than annotating them from scratch. For the cases that do not match any of the conditions listed above, the annotation is done from scratch.

The annotation process is done post-hoc (not in real-time), and therefore Wikidata entries were collected much later, after the whole three years comprised in the dataset were collected. This avoids potential cases of wrong updates on Wikidata impacted by fake reports on Twitter.

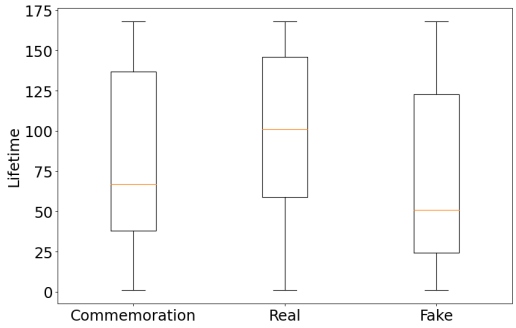

Fig. 1. Lifetimes of death reports grouped by type: commemorations, real deaths and fake deaths.

3.1.5 Final Dataset. The annotation of the 4,007 death reports in our dataset led to the following distribution: 2,301 real deaths, 1,092 commemorations and 614 fake deaths. Table 4 shows the statistics of the dataset. While the categories are imbalanced, this still shows that fake deaths represent a large proportion of all reports (15.3%) and need to be tackled to avoid their diffusion. The skewed distribution of categories presents in turn an additional challenge for the classification task.

The annotation was primarily done by a single annotator, by supervising the automated annotation of the tool described above. The validate the quality of the annotation, a second annotator went through a random subset of 200 death reports, achieving a high inter-rater agreement measured with a Cohen's Kappa of 0.982 [8].

Veracity	Instances	Tweets
Real	2,301	9,131,976
Commemoration	1,092	526,588
Fake	614	643,432
Total	4,007	10,301,996

Table 4. Distribution of labels and tweets in the dataset.

We look at the lifetime (Figure 1) and the number of tweets (Figure 2) of different kinds of death reports.[4] Interestingly, we can see that fake reports have a tendency to last shorter and have fewer tweets posted; still, the median fake reports lasts for about 50 hours. This can be of interest for a behavioural analysis comparing hoaxes and real reports, however it is of little help for an early hoax detection system if we aim to detect hoaxes within a short time after first being posted. Commemorations also have a tendency to last shorter than real deaths, perhaps understandably as the emotional impact of a commemoration is expect to be lower than that of a recent death.

The death reports with the highest number of tweets posted, by accuracy of report, include:

- **Real deaths:** Robin Williams (1.58M tweets), Paul Walker (1.04M), Nelson Mandela (939K), Whitney Houston (462K), Neil Armstrong (324K), Maya Angelou (203K), Cory Monteith (198K), Casey Kasem (139K), Philip Seymour Hoffman (122K), Jenni Rivera (118K).

[4]Note that we only preserve up to 7 days of tweets associated with a death report, as we do not need more for the purposes of our research on early detection of hoaxes. Death reports that may have lasted longer are therefore truncated to 168 hours (7 days) in Figure 1.

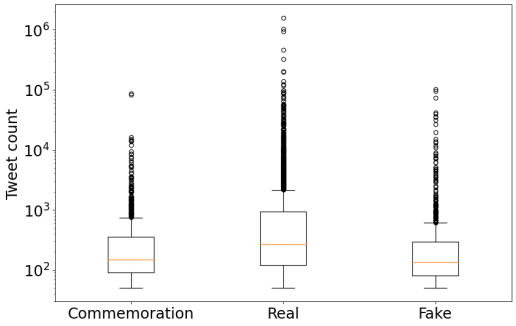

Fig. 2. Tweet counts of death reports grouped by type: commemorations, real deaths and fake deaths.

- **Commemorations:** James Avery (86K tweets), John Lennon (83K), Kurt Cobain (16K), Steve Irwin (15K), Jesus Christ (15K), Eric Garner (14K), Sean Taylor (14K), Gary Speed (12K), George Best (9K), Helen Martin (8K).
- **Death hoaxes:** Megan Fox (101K tweets), Lady Gaga (95K), Chris Brown (73K), Margaret Thatcher (53K), Taylor Swift (40K), Justin Bieber (36K), Eddie Murphy (35K), Channing Tatum (32K), Rowan Atkinson (27K), Ricardo Arjona (20K).

The people who were most repeatedly killed off through death hoaxes include: Justin Bieber (13 times), Soulja Boy (11), Chris Brown (10), Lady Gaga (9), Nicki Minaj (9), Taylor Swift (8), 50 Cent (7), Chuck Norris (7), Eddie Murphy (7), Lebron James (7).

The following are examples of tweets associated with a death hoax.

```
#1: RIP Cesar Millan #ripcesarmillan #dogwhisperer
#2: RIP CESAR MILLAN - You did so much wunnerful fings fah doggiez!
#3: RIP Cesar Millan. I'm sure u will be missed by alot of people, especially the dog lovers
#4: RIP Cesar Millan, gone to soon. Hope you run free over rainbow bridge with all the dogs. #dogwhisperer
#5: We will miss you Cesar we love your show RIP Cesar Millan
#6: So young!! Ugh. RIP Cesar Millan, aka The Dog Whisperer. Very Sad.
```

The hoax went viral with tweets originating from multiple sources and with tweets with very different textual content. Later tweets in the timeline of this viral timeline of tweets start warning others that the story is a hoax: *It's hoax, eat that!*. Despite later correction by Twitter users, the story went viral in what one may consider a credible story, especially if they do not check additional sources for verification.

3.2 Hoax Detection

In this section we provide details of the features and experiment settings that we use for our work.

3.2.1 Classification Features. We use three different types of features, including two features that are widely used in previous work (social features and textual features), as well as our proposed class-specific word representations. Additionally, we propose two different combinations of those features. To simulate the task of early detection of hoaxes, we perform experiments at different points in time. Experiments performed in time t will generate the features only from tweets posted before that time. The feature sets we use for the experiments are as follows:

- **Baseline 1 – Social features (social):** We use a set of 16 features that refer to the reputation of the users participating in a report and to diffusion patterns. Please see the appendix for more details of these features.
- **Baseline 2 – Textual features using word embeddings (w2v):** As a state-of-the-art word representation approach, we use Word2Vec embeddings [32] to represent the content of the tweets associated with a report. The model we use for the embeddings was trained from the entire collection of tweets in the training set, i.e. all the 2012 and 2013 tweets. We represent each tweet as the average of the embeddings for each word, and finally get the average of all tweets.
- **Baseline 3 – Google's word embedding model (gw2v):** As one of the standard and most commonly used word embedding models, we use Google's Word2Vec model with 300 dimensions trained from Google News data[5] as a baseline for comparison with our model.
- **Baseline 4 – Textual features using sentence embeddings (infersent):** InferSent [9] is a state-of-the-art method for semantic representation of sentences into embeddings. Beyond word embeddings, which consider each word as a separate element and ignore aspects like the order and importance of words in a sentence, sentence embeddings methods like InferSent can capture syntactic features. InferSent is a method that learns the structure of sentences from a large natural language inference corpus, the Stanford Natural Language Inference (SNLI) corpus [4], by using a Bi-LSTM encoder.
- **Class-specific word representations (multiw2v):** The same word can have different meanings depending on the category in which it is used. For instance, 'RIP' usually refers to 'Rest In Peace' or 'Requiescat In Pace' when it is used along with a real death, but it can mean 'Really Inspiring Person' when used as a hoax; this is however implicit.[6] This can be hard to distinguish even for humans as the word is exactly the same, but it can be modelled differently using class-specific word embeddings. Provided that we have large-scale training data, we propose to train different word embedding models for each class, so that each model learns the vocabulary of that class. We build three different collections from our training set, each belonging to tweets from one of the categories, and train a separate word embedding model from each of the three collection, so that we have a word embedding for *real* reports, another one for *fake* reports and a third one for *commemorating* reports. Having three different word embedding models (real, fake, commemoration), we then create three different vectors, each of which is created as above, however using a different word embedding model. Finally, we combine all three vectors by concatenating them into a single vector. Our proposed model, which we call *multiw2v*, enables characterisation of reports with respect to each class in the dataset.

We also test combinations of social and different textual features, including word embeddings (social+w2v), sentence embeddings (social+infersent) and class-specific word representations (social+multiw2v).

3.2.2 Experiment Settings. Given that the objective of our experimentation is to find out what features perform best for early detection of hoaxes, assessing the performance of our proposed class-specific word representations, we first tested different classifiers: Support Vector Machines, Random Forests, Logistic Regression, Multi-layer Perceptron, Gaussian Processes and Naive Bayes. We found the Logistic Regression classifier [41] to perform substantially better than the rest of the classifiers, and so for the sake of clarity and space we show results for this classifier in the rest of this article.

[5] https://code.google.com/archive/p/word2vec/

[6] Note that tweets using the 'RIP' keyword all look the same. It is only later when users clarify that they were joking and using the keyword 'RIP' to refer to 'Really Inspiring Person'. Here we mean that class-specific word embeddings can model the keyword 'RIP' differently for real and fake deaths if their context varies.

Additional results for the rest of the classifiers are given in the appendices. We use the implementation of the logistic regression classifier in scikit-learn [39],[7] with the following parameters:

```
solver='liblinear', multi_class='ovr', fit_intercept=True,
intercept_scaling=0.0001, C=0.6, class_weight='balanced'
```

These parameters were determined empirically by testing a range of different parameters on 10-fold cross-validation experiments on held-out parts of the training data. We tested all possible values for categorical parameters, whereas a wide range of values were tested for the numerical parameters, keeping the best-performing parameters in each case and re-testing with nearby values. The high performance computing infrastructure provided by our university was employed for all the experiments.[8]

For the experimentation, We use the first two years (2012 and 2013) for training and the last year (2014) for testing. With this we avoid mixing data from overlapping periods in the training and test sets, and also other cases like having newer data in the training set than in the test set (e.g. if a person died in 2014, we avoid having the real 2014 death in the training set and a fake 2013 death in the test set)[9]. In addition, using old data for training and new data for testing allows simulating a more realistic scenario. Despite having static sets for training and test, we run 10-fold cross-validation experiments with different subsets of the training data. We opted for doing 10-fold cross-validation to enable generalisation of the results and to avoid skewed results affected by a specific training set.

We report performance scores of different classifiers using macroaveraged F1 scores, i.e. averaged F1 scores for the three categories, where the F1 score for a category equates to the harmonic mean between the precision and the recall.

4 RESULTS AND DISCUSSION

We first present a comparison of the different features under study, delving into results by category. Then, we explore the use of sliding windows for the classification.

4.1 Comparison of Features

We first compare the sets of features and combinations of features we described above. We show results for classification experiments in different points in time including 0 (only the first tweet posted), 5, 10, 15, 30, 60, 120, 180 and 300 minutes. This allows us to explore the ability to perform accurate classification early on in the first few minutes, as well as to analyse how much the classifier's performance can improve as time goes on up to 5 hours.

Table 5 shows the results comparing performance of different features. We observe that the approaches using our proposed method for class-specific word representations (multiw2v) perform better than the rest, including the use of standard word embeddings (w2v and gw2v) as well as sentence embeddings (infersent). While social features alone perform poorly, they are actually beneficial when they are combined with the multiw2v features. We see that the combination of social+multiw2v consistently outperforms the sole use of multiw2v features, however this improvement is especially noticeable for later points in time, as the social features become more beneficial with more tweets observed over time; i.e. when the social trend develops. For very early detection of hoaxes, both multiw2v and social+multiw2v perform similarly, with a slightly better performance for the latter. While it is possible to have fairly accurate classification having only observed the first tweet (.669), it is worthwhile delaying the prediction for 2 to 10 minutes to achieve an improved performance (0.696 and 0.716). It is only in later stages in the diffusion of hoaxes, after 5 hours, that the

[7] http://scikit-learn.org/. We use scikit-learn 0.22.2 on Python 3.6.3.
[8] https://docs.hpc.qmul.ac.uk/
[9] This happened for instance with Nelson Mandela, who was killed off multiple times throughout 2012 and 2013 before he actually died in December 2013

combination of social and InferSent features manages to perform slightly better than the social+multiw2v features; this is, however, not ideal for early detection of hoaxes, where we are especially interested in performance results for the early stages of the stream. For earlier stages of the stream, multiw2v-based features clearly outperform InferSent and the other baselines.

	0	1'	2'	5'	10'	15'	30'	60'	120'	300'
social	.427	.495	.509	.510	.510	.528	.535	.577	.594	.591
w2v	.641	*.655*	.658	.663	.667	.670	.680	.696	.699	.698
social+w2v	.612	.634	.661	.671	.671	.677	.675	.709	.709	.724
gw2v	.556	.565	.574	.608	.612	.618	.623	.645	.648	.664
social+gw2v	.569	.590	.599	.616	.633	.647	.663	.679	.688	.686
infersent	.637	.640	.653	.664	.683	.681	.697	.722	.734	.759
social+infersent	*.643*	*.655*	*.670*	*.678*	*.691*	*.688*	*.698*	*.731*	*.748*	**.767**
multiw2v*	**.669**	.676	.691	.703	.714	.722	.723	.721	.738	.741
social+multiw2v*	.647	**.677**‡	**.696**‡	**.707**‡	**.716**‡	**.725**‡	**.724**†	**.744**†	**.752**	*.748*

Table 5. Comparison of features for early detection of hoaxes. Proposed methods indicated with a star (*). Best method highlighted in bold and second best method for different types of features highlighted in italic. ‡: statistically significant at $p < .01$, †: statistically significant at $p < .05$.

4.2 Effect of the Size of the Word Embedding Model

	0	1'	2'	5'	10'	15'	30'	60'	120'	300'
w2v300	.641	.655	.658	.663	.667	.670	**.680**	**.696**	**.699**	.698
w2v600	.631	.643	.656	.657	.658	.663	.669	.681	.690	.693
w2v900	.640	.658	**.667**	**.672**	**.668**	**.679**	.679	.695	**.699**	**.706**
w2v4096	**.642**	**.664**	.663	.667	.666	.671	.677	.695	.698	.697
social+w2v300	.612	.634	.661	.671	.671	.677	.675	**.709**	**.709**	**.724**
social+w2v600	.620	**.654**	**.669**	.675	.670	**.682**	.685	.707	.707	.712
social+w2v900	**.621**	.646	.660	**.680**	**.672**	.681	**.688**	.698	.705	.715
social+w2v4096	.614	.649	**.669**	.675	.668	.681	.683	.697	.702	.713

Table 6. Comparison of performance by using word2vec models of varying dimensionality, including 300 (original w2v above), 600, 900 (equivalent to multiw2v) and 4096 (equivalent to InferSent).

In an additional set of experiments, we aim to determine the extent to which the number of dimensions in the embedding model can impact performance. The motivation behind this experimentation is that the original word2vec model (w2v) creates vectors with 300 dimensions, class-specific embeddings (multiw2v) create vectors with 900 dimensions, and infersent creates vectors with 4096 dimensions. Hence, we perform additional experiments with word2vec models trained with different dimensionalities, which can help us determine if the improvement achieved by multiw2v comes because of the different methodology used or simply because of the higher dimensionality.

Table 6 shows results for word2vec models of dimensionalities of 300, 600, 900 and 4096, both on their own and combined with social features. We can observe that variation in performance is marginal and shows that higher

dimensionality does not lead to improved performance. While this marginal improvement fluctuates slightly, we can observe that on occasions even the use of 300 dimensions can outperform bigger models with 4096 dimensions (e.g. after 10, 30 or 60 minutes). These results demonstrate the potential of multiw2v to provide substantial improvements thanks to leveraging class-specific embeddings and not because of the larger dimensionality.

4.3 Using Sliding Windows

We now experiment with the use of sliding windows for the classification [10]. With sliding windows, we can choose to make use of all the tweets posted so far for a report at time t to classify it, or we can instead make use of a smaller window that only uses the last bit. The motivation behind this is that we hypothesise that Twitter users will show a self-correcting behaviour, potentially being mistaken about the truth of a report in the very early stages, but later correcting themselves as new evidence or more sources are available related to the report. We experiment with different sliding windows by using different percentages. For each percentage, we consider the tweets posted within that fraction of time, counting from the end: $w = \{t - (t - t_0) * p, t\}$, where w is the window comprised between: (1) the current time t minus the percentage p of time between the current time and the time of the first tweet was posted, and (2) the current time.

Table 7 shows the results of using different time windows: 0.1, 0.25, 0.5, 0.75 and 1.0. We use the *social+multiw2v* as the best performing features here for the analysis. With these results we observe that the use of sliding windows is not useful, and that it is much better to use all the tweets associated with a report than the last few. While we do observe that it is better to keep including new tweets as time goes on, which leads to performance gains, we also see that it is important to include all the tweets from the very beginning. Note that results for $t = 0$ are the same in all cases as the use of a sliding window does not have an effect in this case. These results do not support our hypothesis of a self-correcting behaviour happening among users; while some users may possibly correct themselves, there is no sufficient impact on the model to improve performance, hence not validating our hypothesis. Note, however, that we cannot reject the hypothesis for not having investigated it in detail; our experiments lead to the conclusion that the way we modelled this potential self-correcting behaviour leads to performance drop.

window	0	1'	2'	5'	10'	15'	30'	60'	120'	300'
0.1	**.647**	.385	.399	.413	.423	.442	.452	.459	.466	.514
0.25	**.647**	.422	.468	.476	.478	.519	.522	.547	.582	.617
0.5	**.647**	.228	.284	.369	.537	.544	.575	.589	.642	.673
0.75	**.647**	.253	.319	.396	.554	.580	.598	.626	.671	.718
1.0	**.647**	**.677**	**.696**	**.707**	**.716**	**.725**	**.724**	**.744**	**.752**	**.748**

Table 7. Results using sliding windows for early detection of hoaxes, using the best performing set of features (social+multiw2v).

For more results on the impact of sliding windows using other baseline classifiers, please refer to Appendix B.

4.4 Effect of the Size of the Training Data

We now analyse the effect of using different sizes of the training set for the experimentation. We have two years' worth of tweets in the training set, but here we are interested in exploring if we could achieve comparable results by using less training data, which would alleviate the need for having to collect more data prior to running the classifier. We analyse the use of eight different sizes of training sets, with a step size of 3 months between them, i.e. 3, 6, 9, 12, 15, 18,

21 and 24. When we use a number N of months in our training data, we are taking the first N months starting from the beginning of our dataset in January 2012, e.g. 3 months includes January, February and March 2012.

Table 8 shows the results for the use of training sets of different sizes. We observe substantial improvements for the smaller numbers of months, but these improvements become much smaller as we have more training data. Improvements are smaller after we have 12 months of data, but they still keep improving to a lesser extent. It is much later, after the 21st month, that the performance results start to plateau. Differences between using 21 and 24 months are very small, which suggests that it is the optimal result we can get by using this approach. There are, in fact, cases where the classifier performs even better with 21 months of training data than with 24, especially for very early detection of hoaxes for small values of t.

months	0	1'	2'	5'	10'	15'	30'	60'	120'	300'
3	.533	.560	.567	.538	.518	.523	.525	.550	.560	.602
6	.608	.627	.626	.642	.641	.645	.652	.667	.663	.676
9	.632	.634	.644	.641	.648	.665	.664	.686	.692	.691
12	.637	.666	.666	.678	.687	.691	.697	.709	.710	.721
15	.645	.663	.679	.690	.701	.713	.721	.733	.732	.730
18	.643	.668	.683	.695	.708	.714	.718	.731	.735	.733
21	**.649**	.675	.689	.699	**.718**	.720	**.728**	.737	.744	.744
24	.647	**.677**	**.696**	**.707**	.716	**.725**	.724	**.744**	**.752**	**.748**

Table 8. Performance results for early detection of hoaxes by using different sizes of training data, in months, using the best performing set of features (social+multiw2v).

For more results on different sizes of the training data using other baseline classifiers, please refer to Appendix B.

4.5 Effect of the Data Sampling Strategy

In the process of generating our dataset of death reports, we made the decision of setting 50 as the minimum number of tweets that a report would need to reach in order to be included in the dataset. This decision was made for scalability issues and for making sure that we have enough data for each report. To determine the impact of setting 50 as the value for this threshold, here we experiment with two other thresholds, 100 and 150. The objective is to see if we get similar results or instead a different threshold would lead to different conclusions.

Figure 3 shows slight variations as the threshold changes. The use of 'social+multiw2v' and 'social+infersent' features consistently perform better than the other features. While there are occasions where 'social+infersent' can perform best, 'social+multiw2v' generally achieves the best performance, particularly in earlier stages of the reports, i.e. in the first 100 minutes. When the threshold is increased to 150, the performance of 'social+multiw2v' is slightly below 'social+infersent' for the first few minutes only.

5 ANALYSIS OF FEATURES

Figure 4 shows the values for the 16 social features in our experiments, plotted as a timeline showing their values over time per category. Feature values are averaged across all of the instances for a particular category. Some figures show a similar increasing/decreasing tendency of values across categories, which is however affected by the normalisation of values we perform. These figures are especially useful to distinguish the values across categories.

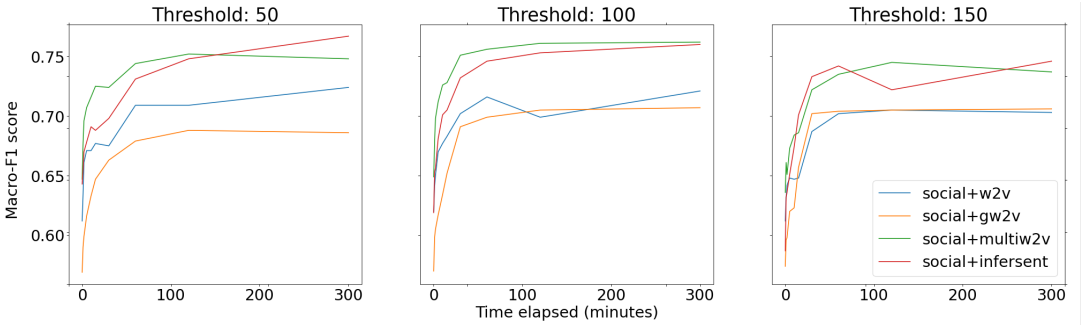

Fig. 3. Performance across different sampling thresholds.

We are particularly interested in looking at the cases where fake deaths (red lines) exhibit very different values with respect to real and commemorative deaths (blue and green lines). Some interesting findings we observe from this analysis:

- The ratio of distinct users (user ratio) is lower for death hoaxes than it is for the other categories, showing that the number of users who participate in fake stories tends to be lower.
- This differs, however, from the the ratio of distinct retweeting users (retweeting user ratio). Here we observe instead that the number of distinct users retweeting death hoaxes tends to be higher than for the other two categories, hence showing that death hoaxes are initiated by a few but retweeted by many.
- Tweets associated with death hoaxes tend to be shorter (tweet length) and have fewer words (token ratio), possibly indicating the provision of less context or evidence associated with the report. This ties in with the finding that death hoaxes are less likely to provide links (link ratio), likely owing to the lack of news articles covering those.
- Death hoaxes tend to spark more questions (question ratio), which may indicate more skepticism from participating users.
- It is the case of commemorations that is more often accompanied with pictures (picture ratio), than for real or fake deaths. This indicates a higher likelihood to commemorate people from the past with memorable pictures.

6 DISCUSSION AND CONCLUSION

We have introduced a novel approach for semi-automated generation of annotated social media datasets made of celebrity death reports for veracity classification. Different from previous work, our approach does not need to collect true and false stories using different approaches, and consequently enables experimentation in a realistic scenario with a realistic ratio of false stories. Our semi-automated approach consists in leveraging the Wikidata knowledge base, with which we can easily verify if celebrity death reports circulating in social media refer to people who have actually died or are instead made up reports. Following this process, we have produced a dataset comprising 4,007 different death reports, which include over 13 million tweets, and have a ratio of 15% false stories.

The generation of this dataset has also enabled us to run experiments for early hoax detection from social media, which we have experimented for very early detection within minutes of the first report. Taking advantage of the large-scale of our dataset, we have experimented using class-specific representations of word embeddings. This approach

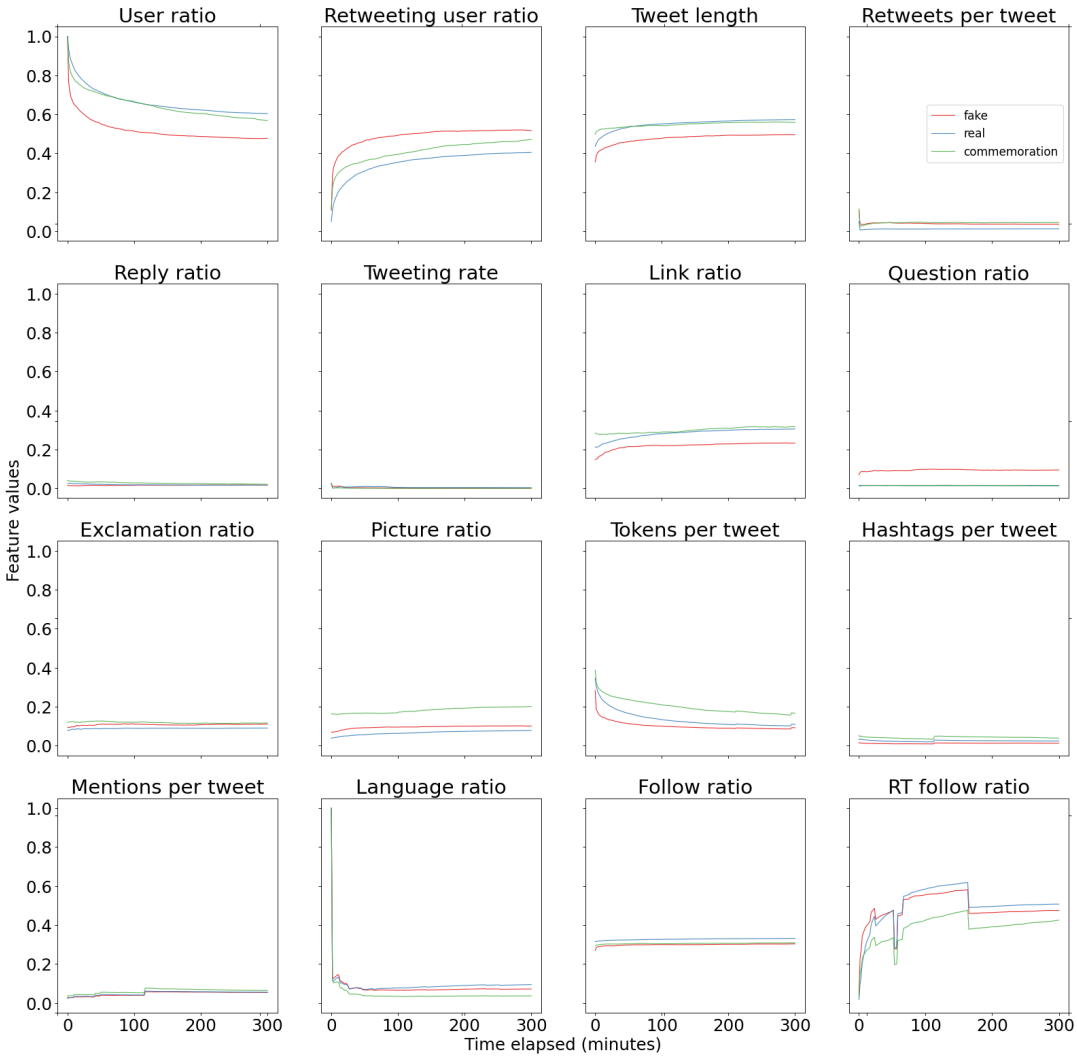

Fig. 4. Temporal visualisation of social feature values, comparing across labels: real, commemoration and fake.

has proven to clearly outperform the use of a single model of word embeddings for the entire dataset. Our approach achieves competitive results for detection of hoaxes within the first 2 to 10 minutes, with F1 scores close to 72% within 10 minutes. This method based on class-specific embeddings for early detection of hoaxes outperforms state-of-the-art methods for word representation, including word embeddings (w2v) and sentence embeddings (infersent). Different from the latter, our proposed method, multiw2v, is able to leverage the class labels from a large-scale, annotated dataset, to learn different meanings of words across categories, e.g. RIP meaning 'Rest in Peace' or 'Really Inspiring Person', depending on the type of report. While we have tested the multiw2v word representation approach for the hoax detection task, it is directly applicable to any other classification task where a large-scale annotated dataset is available for building the multiw2v model, for instance thanks to the availability of distantly supervised datasets.

With further experimentation, we have observed that the use of sliding windows, where the most recent tweets are considered for the classification task, is not helpful in this task, and instead using the entire timeline of tweets is better. We have observed that the larger the sliding window, the better is our system's performance, with optimal results for a sliding window covering 100% of the stream, i.e. the equivalent of not having a sliding window. Finally, we have explored the effect of having different sizes of training sets, showing that performance results start to plateau after 21 months of training data and more training data may not necessarily lead to improved results.

The dataset and the word embedding models developed in this work are publicly available,[10] enabling further research in this much needed research area using a benchmark dataset.

Our data collection, annotation and experimentation is limited to a specific kind of hoax triggered by death reports. Our motivation to focus on this kind of reports was both their prominence in social media and the need to tackle them, as well as the possibility of modelling the problem by leveraging names of notable people from Wikipedia. Hence this led to the development of a novel method to develop a large-scale dataset to tackle hoaxes, while being restricted to this specific kind of hoaxes. This has the limitation of its direct applicability to broader types of hoaxes, which this work does not cover and is left for future work. This, in turn, should be taken into account when interpreting the findings of this work, whose generalisation to other kinds of hoaxes needs further investigation.

Our plans for future work include experimentation with other events that can be linked to Wikidata or other knowledge bases, beyond death reports, such as resignation of public figures, numbers of casualties reported for emergency events, or other factual claims.

ACKNOWLEDGMENTS

This research utilised Queen Mary's Apocrita HPC facility, supported by QMUL Research-IT. http://doi.org/10.5281/zenodo.438045

REFERENCES

[1] Sarah A Alkhodair, Steven HH Ding, Benjamin CM Fung, and Junqiang Liu. 2020. Detecting breaking news rumors of emerging topics in social media. *Information Processing & Management* 57, 2 (2020), 102018.

[2] Hunt Allcott and Matthew Gentzkow. 2017. *Social media and fake news in the 2016 election.* Technical Report. National Bureau of Economic Research.

[3] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research* 3, Feb (2003), 1137–1155.

[4] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*. Association for Computational Linguistics (ACL).

[5] Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-based n-gram models of natural language. *Computational linguistics* 18, 4 (1992), 467–479.

[6] Axel Bruns, Tim Highfield, and Rebecca Ann Lind. 2012. Blogs, Twitter, and breaking news: The produsage of citizen journalism. *Produsing theory in a digital world: The intersection of audiences and production in contemporary theory* 80, 2012 (2012), 15–32.

[7] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2011. Information credibility on twitter. In *Proceedings of the 20th international conference on World wide web*. ACM, 675–684.

[8] Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement* 20, 1 (1960), 37–46.

[9] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 670–680.

[10] Mayur Datar, Aristides Gionis, Piotr Indyk, and Rajeev Motwani. 2002. Maintaining stream statistics over sliding windows. *SIAM journal on computing* 31, 6 (2002), 1794–1813.

[11] Leon Derczynski, Kalina Bontcheva, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Arkaitz Zubiaga. 2017. SemEval-2017 Task 8: RumourEval: Determining rumour veracity and support for rumours. In *Proceedings of SemEval*. ACL, 69–76.

[12] Nicholas Diakopoulos, Munmun De Choudhury, and Mor Naaman. 2012. Finding and assessing social media information sources in the context of journalism. In *Proceedings of CHI*. ACM, 2451–2460.

[10] https://figshare.com/articles/Twitter_Death_Hoaxes_dataset/5688811

[13] Sebastian Dungs, Ahmet Aker, Norbert Fuhr, and Kalina Bontcheva. 2018. Can rumour stance alone predict veracity?. In *Proceedings of the 27th International Conference on Computational Linguistics*. 3360–3370.

[14] Paolo Gerbaudo. 2018. *Tweets and the streets: Social media and contemporary activism*. Pluto Press.

[15] Fantahun Bogale Gereme and William Zhu. 2019. Early Detection of Fake News" Before It Flies High". In *Proceedings of the 2nd International Conference on Big Data Technologies*. 142–148.

[16] Jennifer Golbeck, Matthew Mauriello, Brooke Auxier, Keval H Bhanushali, Christopher Bonk, Mohamed Amine Bouzaghrane, Cody Buntain, Riya Chanduka, Paul Cheakalos, Jennine B Everett, et al. 2018. Fake news vs satire: A dataset and analysis. In *Proceedings of the 10th ACM Conference on Web Science*. 17–21.

[17] Genevieve Gorrell, Kalina Bontcheva, Leon Derczynski, Elena Kochkina, Maria Liakata, and Arkaitz Zubiaga. 2019. RumourEval 2019: Determining Rumour Veracity and Support for Rumours. *Proceedings of SemEval* (2019), 845–854.

[18] Jeffrey Gottfried and Elisa Shearer. 2016. *News Use Across Social Media Platforms 2016*. Technical Report. Pew Research Center.

[19] Aditi Gupta, Ponnurangam Kumaraguru, Carlos Castillo, and Patrick Meier. 2014. Tweetcred: Real-time credibility assessment of content on twitter. In *International Conference on Social Informatics*. Springer, 228–243.

[20] Alfred Hermida. 2012. Tweets and truth: Journalism as a discipline of collaborative verification. *Journalism Practice* 6, 5-6 (2012), 659–668.

[21] Zhiwei Jin, Juan Cao, Yongdong Zhang, and Jiebo Luo. 2016. News Verification by Exploiting Conflicting Social Viewpoints in Microblogs.. In *AAAI*. 2972–2978.

[22] Sarthak Jindal, Mayank Vatsa, and Richa Singh. 2019. Newsbag: a benchmark dataset for fake news detection. (2019).

[23] Terry Koo, Xavier Carreras Pérez, and Michael Collins. 2008. Simple semi-supervised dependency parsing. In *Proceedings of ACL*. 595–603.

[24] Srijan Kumar, Robert West, and Jure Leskovec. 2016. Disinformation on the web: Impact, characteristics, and detection of wikipedia hoaxes. In *Proceedings of WWW*. 591–602.

[25] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. 2010. What is Twitter, a social network or a news media?. In *Proceedings of WWW*. ACM, 591–600.

[26] Sejeong Kwon, Meeyoung Cha, and Kyomin Jung. 2017. Rumor detection over varying time windows. *PloS one* 12, 1 (2017), e0168344.

[27] Xiaomo Liu, Armineh Nourbakhsh, Quanzhi Li, Rui Fang, and Sameena Shah. 2015. Real-time rumor debunking on twitter. In *Proceedings of CIKM*. ACM, 1867–1870.

[28] Yang Liu and Yi-Fang Brook Wu. 2018. Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[29] Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam-Fai Wong, and Meeyoung Cha. 2016. Detecting Rumors from Microblogs with Recurrent Neural Networks.. In *IJCAI*. 3818–3824.

[30] Curtis Daniel MacDougall. 1958. *Hoaxes*. Vol. 465. Dover Pubns.

[31] Filippo Menczer. 2016. The spread of misinformation in social media. In *Proceedings of WWW*. 717–717.

[32] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[33] Tanushree Mitra and Eric Gilbert. 2015. Credbank: A large-scale social media corpus with associated credibility annotations. In *Ninth International AAAI Conference on Web and Social Media*.

[34] João Moreno and Graça Bressan. 2019. FACTCK. BR: a new dataset to study fake news. In *Proceedings of the 25th Brazillian Symposium on Multimedia and the Web*. 525–527.

[35] Meredith Ringel Morris, Scott Counts, Asta Roseway, Aaron Hoff, and Julia Schwarz. 2012. Tweeting is believing?: understanding microblog credibility perceptions. In *Proceedings of CSCW*. ACM, 441–450.

[36] Kai Nakamura, Sharon Levy, and William Yang Wang. 2019. r/Fakeddit: A New Multimodal Benchmark Dataset for Fine-grained Fake News Detection. *arXiv preprint arXiv:1911.03854* (2019).

[37] Robert Nares. 1822. *A Glossary: Or, Collection of Words, Phrases, Names, and Allusions to Customs, Proverbs, &c., which Have Been Thought to Require Illustration, in the Works of English Authors, Particularly Shakespeare, and His Contemporaries...* R. Triphook.

[38] Jeppe Nørregaard, Benjamin D Horne, and Sibel Adalı. 2019. NELA-GT-2018: A large multi-labelled news dataset for the study of misinformation in news articles. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 13. 630–638.

[39] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[40] Vahed Qazvinian, Emily Rosengren, Dragomir R Radev, and Qiaozhu Mei. 2011. Rumor has it: Identifying misinformation in microblogs. In *Proceedings of EMNLP*. 1589–1599.

[41] Adwait Ratnaparkhi. 1997. A simple introduction to maximum entropy models for natural language processing. *IRCS Tech. Reports Series* (1997), 81.

[42] Julio CS Reis, André Correia, Fabrício Murai, Adriano Veloso, Fabrício Benevenuto, and Erik Cambria. 2019. Supervised Learning for Fake News Detection. *IEEE Intelligent Systems* 34, 2 (2019), 76–81.

[43] Fatima K Abu Salem, Roaa Al Feel, Shady Elbassuoni, Mohamad Jaber, and May Farah. 2019. FA-KES: a fake news dataset around the Syrian war. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 13. 573–582.

[44] Justin Sampson, Fred Morstatter, Liang Wu, and Huan Liu. 2016. Leveraging the implicit structure within social media for emergent rumor detection. In *Proceedings of CIKM*. ACM, 2377–2382.

[45] Jagan Sankaranarayanan, Hanan Samet, Benjamin E Teitler, Michael D Lieberman, and Jon Sperling. 2009. Twitterstand: news in tweets. In *Proceedings of SIGSPATIAL*. ACM, 42–51.

[46] Chengcheng Shao, Giovanni Luca Ciampaglia, Onur Varol, Kai-Cheng Yang, Alessandro Flammini, and Filippo Menczer. 2018. The spread of low-credibility content by social bots. *Nature communications* 9, 1 (2018), 4787.

[47] Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu. 2018. FakeNewsNet: A Data Repository with News Content, Social Context and Spatialtemporal Information for Studying Fake News on Social Media. *arXiv preprint arXiv:1809.01286* (2018).

[48] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake News Detection on Social Media: A Data Mining Perspective. *ACM SIGKDD Explorations Newsletter* 19, 1 (2017), 22–36.

[49] Eugenio Tacchini, Gabriele Ballarin, Marco L Della Vedova, Stefano Moret, and Luca de Alfaro. 2017. Some Like it Hoax: Automated Fake News Detection in Social Networks. In *Proceedings of the Second Workshop on Data Science for Social Good*.

[50] Tetsuro Takahashi and Nobuyuki Igata. 2012. Rumor detection on twitter. In *Proceedings of SCIS*. IEEE, 452–457.

[51] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a Large-scale Dataset for Fact Extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 809–819.

[52] Sebastian Tschiatschek, Adish Singla, Manuel Gomez Rodriguez, Arpit Merchant, and Andreas Krause. 2018. Fake news detection in social networks via crowd signals. In *Companion Proceedings of the The Web Conference 2018*. International World Wide Web Conferences Steering Committee, 517–524.

[53] Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of ACL*. 384–394.

[54] Inna Vogel and Peter Jiang. 2019. Fake News Detection with the New German Dataset "GermanFakeNC". In *International Conference on Theory and Practice of Digital Libraries*. Springer, 288–295.

[55] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85.

[56] William Yang Wang. 2017. "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 422–426.

[57] Yaqing Wang, Fenglong Ma, Zhiwei Jin, Ye Yuan, Guangxu Xun, Kishlay Jha, Lu Su, and Jing Gao. 2018. EANN: Event adversarial neural networks for multi-modal fake news detection. In *Proceedings of the 24th acm sigkdd international conference on knowledge discovery & data mining*. 849–857.

[58] Fan Yang, Yang Liu, Xiaohui Yu, and Min Yang. 2012. Automatic detection of rumor on sina weibo. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*. ACM, 13.

[59] Arefeh Yavary, Hedieh Sajedi, and Mohammad Saniee Abadeh. 2020. Information verification in social networks based on user feedback and news agencies. *Social Network Analysis and Mining* 10, 1 (2020), 2.

[60] Xichen Zhang and Ali A Ghorbani. 2020. An overview of online fake news: Characterization, detection, and discussion. *Information Processing & Management* 57, 2 (2020), 102025.

[61] Xinyi Zhou, Atishay Jain, Vir V Phoha, and Reza Zafarani. 2019. Fake news early detection: A theory-driven model. *arXiv preprint arXiv:1904.11679* (2019).

[62] Xinyi Zhou, Reza Zafarani, Kai Shu, and Huan Liu. 2019. Fake news: Fundamental theories, detection strategies and challenges. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. ACM, 836–837.

[63] Arkaitz Zubiaga. 2019. Mining social media for newsgathering: A review. *Online Social Networks and Media* 13 (2019), 100049.

[64] Arkaitz Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter. 2018. Detection and resolution of rumours in social media: A survey. *ACM Computing Surveys (CSUR)* 51, 2 (2018), 32.

[65] Arkaitz Zubiaga and Heng Ji. 2014. Tweet, but verify: epistemic study of information verification on Twitter. *Social Network Analysis and Mining* 4, 1 (2014), 1–12.

[66] Arkaitz Zubiaga, Heng Ji, and Kevin Knight. 2013. Curating and contextualizing twitter stories to assist with social newsgathering. In *Proceedings of IUI*. ACM, 213–224.

[67] Arkaitz Zubiaga, Elena Kochkina, Maria Liakata, Rob Procter, and Michal Lukasik. 2016. Stance classification in Rumours as a Sequential Task Exploiting the Tree Structure of Social Media Conversations. In *Proceedings of International Conference on Computational Linguistics, COLING*. 2438–2448.

[68] Arkaitz Zubiaga, Elena Kochkina, Maria Liakata, Rob Procter, Michal Lukasik, Kalina Bontcheva, Trevor Cohn, and Isabelle Augenstein. 2018. Discourse-aware rumour stance classification in social media using sequential classifiers. *Information Processing & Management* 54, 2 (2018), 273–290.

[69] Arkaitz Zubiaga, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Peter Tolmie. 2016. Analysing how people orient to and spread rumours in social media by looking at conversational threads. *PloS one* 11, 3 (2016).

A LIST OF SOCIAL FEATURES

With the social features we create vectors with 16 values, all of which are normalised to be between 0 and 1:

- **User ratio:** Number of unique users divided by the number of tweets.
- **Retweeting user ratio:** Number of unique retweeting users divided by the number of tweets.
- **Tweet length:** Average length of tweets in characters.
- **Retweets per tweet:** Average number of retweets per tweet.
- **Reply ratio:** Number of tweets that are replying to another tweet divided by the number of all tweets.
- **Tweeting rate:** Number of tweets per second.
- **Link ratio:** Number of links found in all tweets divided by the number of tweets.
- **Question ratio:** Number of question marks found in all tweets divided by the number of tweets.
- **Exclamation ratio:** Number of exclamation marks found in all tweets divided by the number of tweets.
- **Picture ratio:** Number of pictures found in all tweets divided by the number of tweets.
- **Tokens per tweet:** Number of (space-separated) tokens found in all tweets divided by the number of tweets.
- **Hashtags per tweets:** Number of unique hashtags found in all tweets divided by the number of tweets.
- **Mentions per tweet:** Number of unique user mentions found in all tweets divided by the number of tweets.
- **Language ratio:** Number of unique languages used in the tweets divided by the number of tweets.
- **Average follow ratio of users:** We compute the average of the follow ratios of all users. The follow ratio of a user is computed as $log_{10}(following)/log_{10}(followers)$.
- **Average follow ratio of retweeting users:** We compute the average of the follow ratios of all the retweeting users.

B ADDITIONAL RESULTS WITH BASELINE CLASSIFIERS

B.1 Use of Sliding Windows with All Classifiers

We compare the impact of different window sizes on the rest of the classifiers that we tested as baselines. Figure 5 shows results for different window sizes for Gaussian Processes, Multi-layer Perceptron, Support Vector Machines (SVM), Random Forest and Naive Bayes, along with Logistic Regression. It shows that the tendency for achieving optimal results using the entire window (1.0) holds for all of the classifiers under study.

B.2 Impact of Training Sizes with All Classifiers

Figure 6 shows the performance of the six different classifiers using 6, 12, 18 and 24 months' worth of data for training. With the exception of the naive bayes classifier showing a very similar performance irrespective of the size of the training data, the rest of the classifiers show a consistent tendency for improving performance as the training data increases. It can be seen, however, that this improvement tends to be larger from 6 to 12 months, with slightly smaller improvements when the training data is augmented to include 18 or 24 months.

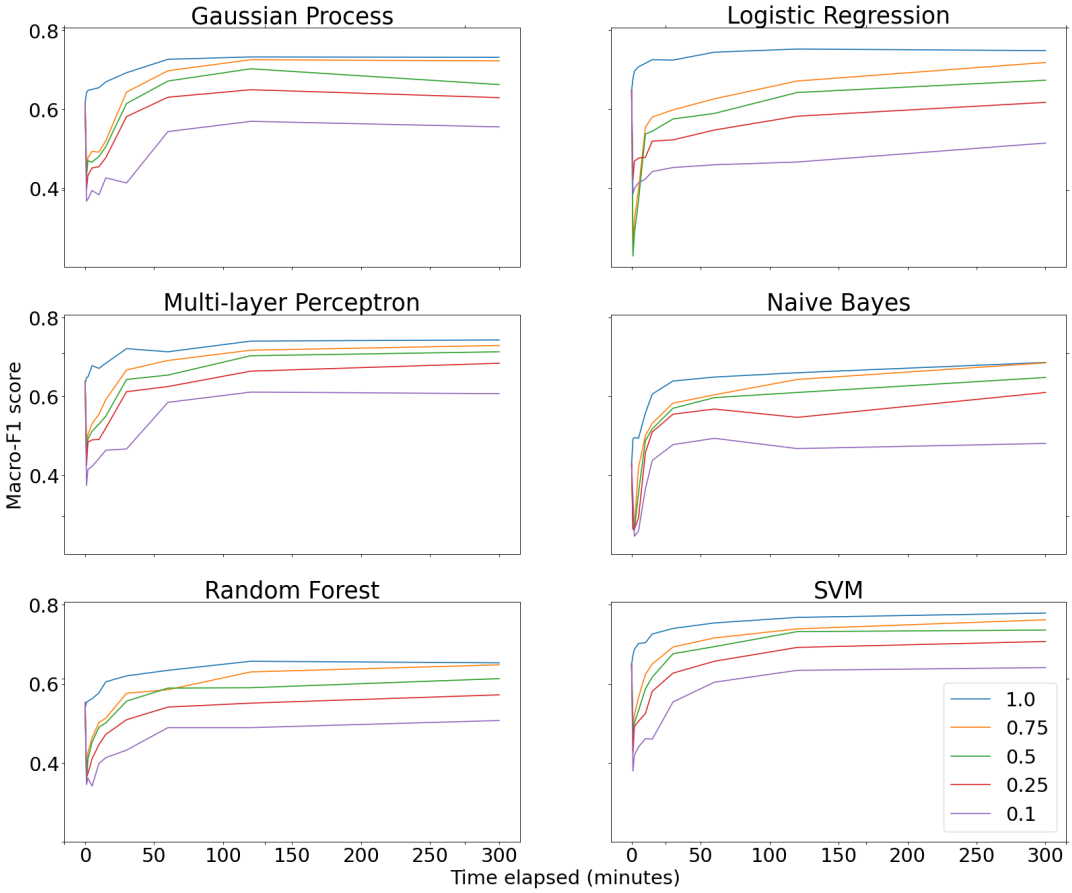

Fig. 5. Performance of different classifiers using different sizes of sliding windows.

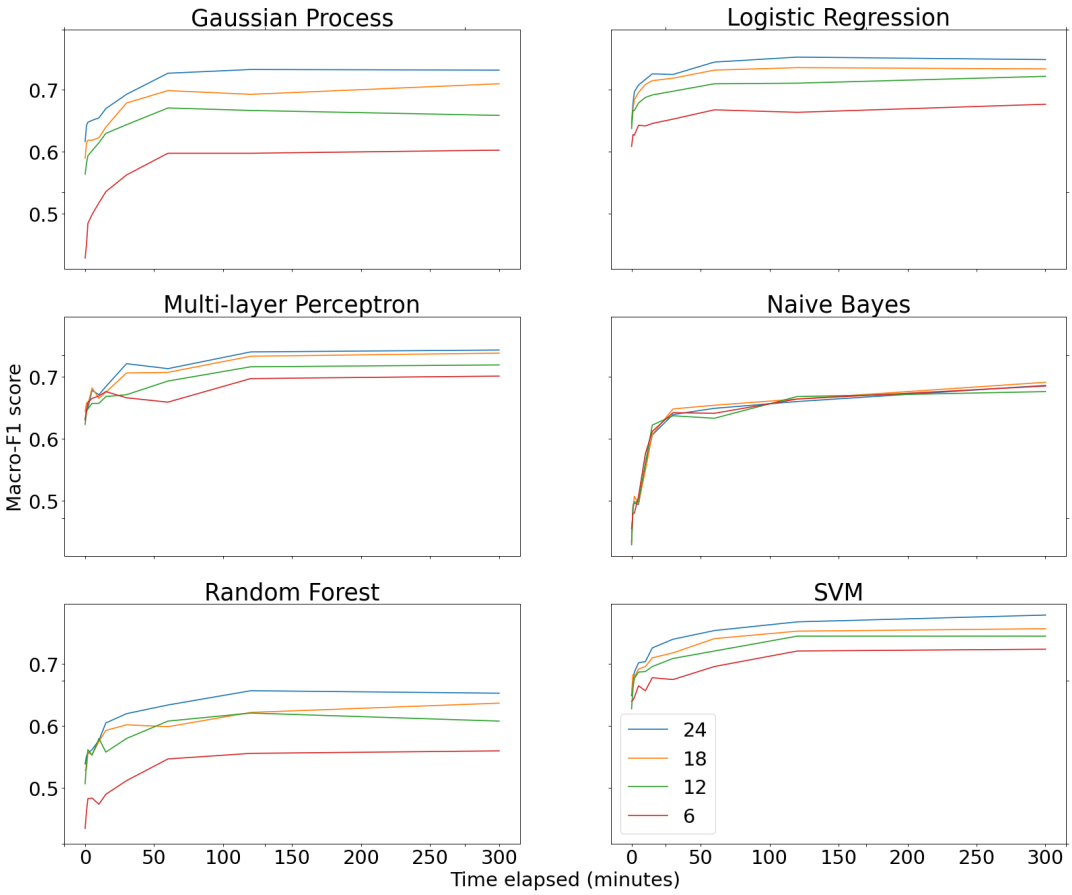

Fig. 6. Performance of different classifiers using different sizes of training data, in months. Figures are limited to 6, 12, 18 and 24 months to avoid saturation of lines and facilitate visualisation.