Check-It: A plugin for Detecting Fake News on the Web

Demetris Paschalides[a], Chrysovalantis Christodoulou[a], Kalia Orphanou[c], Rafael Andreou[a], Alexandros Kornilakis[b], George Pallis[a], Marios D. Dikaiakos[a], Evangelos Markatos[b]

[a] *Computer Science Department, University of Cyprus, Leoforos Panepistimiou 1, Cyprus*
[b] *Computer Science Department, University of Crete, Heraklion 700 13, Greece*
[c] *Open University of Cyprus, Giannou Kranidioti 33, Latsia, Cyprus*

Abstract

The rapid proliferation of misinformation and disinformation on the Internet has brought dire consequences upon societies around the world, fostering extremism, undermining social cohesion and threatening the democratic process. This impact can be attested by recent events like the COVID-19 pandemic and the 2020 US presidential election. The impact of misinformation has been so deep and wide that several authors characterize the present historic period as the "post-truth" era. Many recent efforts seek to contain the proliferation of misinformation by automating the identification of fake news through various techniques that exploit signals derived from linguistic processing of online content, analysis of message diffusion patterns, reputation lists, etc. In this paper we describe the design, implementation of, and experimentation with Check-It, a lightweight, privacy preserving browser plugin that detects fake-news. Check-It combines knowledge extracted from a variety of signals, and outperforms state-of-the-art methods on commonly-used datasets, achieving more than 90% accuracy, as well as a smooth user experience.

Keywords: Fake News Detection, Browser Plugin, Feature Selection, Misinformation, Machine Learning

1. Introduction

The widespread of online social networking and media platforms has changed dramatically the production and consumption of digital information. Any individual equipped with an Internet connection and a social media account can create and circulate content that can reach people at unprecedented speed and scale, without any prior moderation for inaccuracy or inappropriateness. This trend has led to a global misinformation and disinformation crisis with grave consequences for societies around the world, to the extent that many authors named the current historical period as the "post-truth" era. The term "post-truth" was declared as the 2016 international Word of the Year[1] by Oxford Dictionaries. It signifies contexts "*relating to or denoting circumstances in which objective facts are less influential in shaping public opinion than appeals to emotion and personal belief*". It has been extensively used to describe the context of prominent political or social events and phenomena shaped by misinformation, conspiracy theories, and fake news [1, 2], such as Brexit [2], Donald Trump's 2016 US presidential election campaign [3], the ongoing pandemic crisis (COVID-19)[4], and the 2020 United States presidential elections[5].

It is clear that the spread of fake news brought grave effects upon social cohesion and the democratic process [3, 4] and has raised great concern amongst political, media, and academic circles, prompting investigations that seek to identify, analyze and understand this phenomenon and its underlying processes. In the recent research literature, there have been many different approaches for identifying and mitigating misinformation. Although these approaches differ in their choice of algorithmic techniques and their adaptation, they do share common techniques of methodology and deployment. At first, they define *i)* a variety of input signals for their fake news identification component. Such input signals are typical: the reputation of news sources maintained as a form of flag-lists; fact-check annotations, which are produced manually by human editors, and the output of Machine Learning (ML) classification models, which consume information retrieved from online social networks and news articles. Afterward, they proceed with *ii)* packaging and deploying the technique of choice as browser-plugins, which assist users in their daily browsing experience. Such examples are the First Draft

[1] `https://languages.oup.com/word-of-the-year/2016/`
[2] `https://www.bbc.com/news/blogs-trending-48356351`
[3] `https://www.bbc.com/news/world-us-canada-37896753`
[4] `https://www.un.org/en/battling-covid-19-misinformation-hands`
[5] `https://tinyurl.com/y3m9hpfl`

News project CrossCheck[6], B.S. Detector[7], and the NewsGuard[8], which make use of domain flag-lists and source reputations; the TrustedNews[9] and FakerFact[10], which employ Machine Learning (ML) and Deep Learning (DL) algorithms over textual content of articles; and TweetCred[5] which utilizes social network properties to determine the veracity of a post.

However, if we analyze current fake news detection and deployment approaches, two issues are raised: first, the need to consider and combine more signals in the identification process, in order to boost its overall effectiveness. Notably, prior approaches utilize a single input signal (either flag-lists, fact-checks, article content, or social network). To this end, however, we need to come up with more effective signals, and with approaches for combining them efficiently. The second issue is related to the preservation of end-user privacy when assessing visited pages, by not revealing the user's identity and browsing history to any third-party services, in compliance with EU's GDPR policy[11].

To address these issues, we designed and implemented Check-It, a fake news identification system developed as a browser-plugin. Check-It bundles together a series of diverse signals, including flag-lists, similarity matching, and Artificial Intelligence (AI) techniques, making it able to calculate the credibility of a piece of news and successfully warn the reader, whilst securing his/her privacy (GDPR compliant) by working locally on the browser without the need for external communication (i.e. API services). This article substantially extends our previous work [6], where we introduced the Check-it browser plugin, as follows:

- Check-It has been re-designed as a modular software that supports fake news identification based on a variety of signals.
- Check-It has been enhanced by a two-phase feature selection process, which uses L2 regularization and a Genetic Algorithm (GA), to identify

[6]`https://firstdraftnews.org/project/crosscheck/`
[7] `http://bsdetecor.tech`
[8] `http://www.newsguardtech.com/`
[9]`https://trusted-news.com/`
[10]`https://www.fakerfact.org/`
[11]`https://tinyurl.com/yyv6k6np`

a limited number of features that can train successfully a low-resource, light memory Logistic Regression (LR) model. Extensive experiments show that the proposed feature selection method outperforms state-of-the-art alternatives.

- A thorough evaluation and comparison to other state-of-the-art works is conducted with real-world data. Results show that Check-it outperforms existing works, achieving more than 90% accuracy.
- The Check-It plugin[12] is available for the community and can be installed in several browsers (including Google's Chrome, Mozilla Firefox, etc.).

The rest of this work is organized as follows. In Section 2, the related work in the field is presented. In Section 3, we present the Check-It system. Section 4 describes the feature engineering process. Section 5 showcase the plugin and the user flow. In Section 6, we present our experimental setup and the evaluation of the performance of Check-It. In Section 7, the key findings of this work are discussed, and finally, in Section 8, we conclude this paper.

2. Related Work

Prior works on detecting and analyzing misinformation rely on large amounts of annotated data sets to train supervised models. In this context, existing research has focused either on content-based analysis and linguistic styles of fake news articles [7, 8, 9] or propagation-based methods, by studying the behavior of the diffusion of fake news articles in online social networks [10, 11, 12, 13]. There also exist hybrid works that combine both the linguistic and social context signals in more holistic approaches to identify fake news articles [14, 15, 16, 6]. In this section, we present the literature review on different approaches for the above.

2.1. Content-based Fake News Detection

Digging into the content of news articles using Natural Language Processing (NLP) has experimentally proven to be effective in recognizing discrepancies between genuine and forged articles [7, 8, 9]. The potential of NLP and textual content analysis is visible in the work of Potthast et al. [7]. The authors

[12] `https://tinyurl.com/y4tmakjg`

capture linguistic-based features, including specific writing styles and sensational headlines that commonly occur in fake news. They identify writing-style characteristics able to distinguish between articles origin from hyper-partisan and balanced viewpoints, while they observe notable similarities in writing styles of different political orientations (Left and Right-wing extremism). In another work, Horne et al. [8] applied an extensive analysis of the content and title of fake and real news articles. Specifically, they argue that fraudulent news titles contain fewer stop-words and nouns, while they notice more usage of proper-nouns and verb phrases in fake news. Combining the aforementioned works' features and by incorporating several more features extracted from articles' body and headlines, Paschalides et al. [6] constructed an extensive set of 535 linguistic features which were used to train the initial Deep Learning model of the Check-It system.

2.2. Propagation-based Fake News Detection

In addition to news' content, social context-based approaches incorporate features from social media user profiles, post contents, and social networks. For instance, Vosoughi et al. [10] focus on how differently falsehood stories propagate on Twitter, in contrast to real ones. They prove that fraudulent news diffuses significantly faster, deeper, and more broadly than the truth. Moreover, Castillio et al. [11] analyze user's credibility on Twitter based on posts and retweets. The authors show that the automatic credibility assessment on newsworthy messages is possible via post and propagation features. They observe that tweets with credible news are propagated through users with high posting frequency, and with a higher probability of their posts being shared. This observation is also the main intuition behind the Check-It approach in analyzing user behavior in posting fake news articles. In addition, the authors of Jin et al. [12] exploit the users' conflicting viewpoints for verifying the credibility of a news piece. The analysis and verification are applied over a credibility propagation network of tweets that are constructed with both supporting and opposing relations of users/tweets based on the computed viewpoints. The authors showcase the effectiveness of their approach by evaluating an annotated dataset.

2.3. Hybrid Fake News Detection

Despite the numerous efforts for identifying fake news articles, either based on the articles' content or how they propagate in online social mediums, the problem still exists, and the effectiveness of the different approaches is not sufficient. This has urged researchers to utilize a combination of both content-based and social context-based signals [14, 15, 16]. In their work, Ruchansky et al. [14] present a hybrid DL model (CSI) that uses a multi-modal approach, by combining the content of the article, the response of social network users to the article, and the users that promote the article. Following the same intuition, Shu et al. [15] propose the TriFn framework which models tri-relationship for fake news detection. Specifically, the TriFn extracts features from the relationship between publishers, users, and news, using embeddings. The rich knowledge of the tri-relationship offers a significant improvement in the identification of fake news articles over other baseline approaches.

Despite the advantages of the process and the high dimensional feature space, which most of the previously mentioned works have, only a few of them apply feature selection. For instance, Shu et al. [15] use a non-negative matrix factorization (NMF) algorithm for document representation which reduces the dimensionality of the features, while Potthast et al. [7] discard the features that are not worthwhile, manually. Furthermore, some other works [14, 6], rather than performing feature reduction, they use the extensive set of features as input to a deep learning model.

Our work differentiates from the previously mentioned works by the significantly larger amount of features extracted and analyzed in order to obtain a deeper understanding of the article's context. To reduce the noise, we apply the proposed feature selection method which maintains only the most significant features for the identification of fake news. Moreover, in the Check-It plugin, we combined a variety of different signals, such as flag-listing, linguistic features, content similarity, and social network, to support our decision-making and increase the validity of our results.

3. Check-It System

In this section, we introduce the design, architecture, and key features of Check-It. The overall system has four main components which are depicted in Figure 1: a) **Flag-list Matcher** matches the sources of news articles to Known Fake News Domains and Fact Check Sources; b) **Fact Check Similarity** compares a news article against Known Fact Checked Articles labeled as fake from Fact-Checking organizations; c) **Online Social Network User**

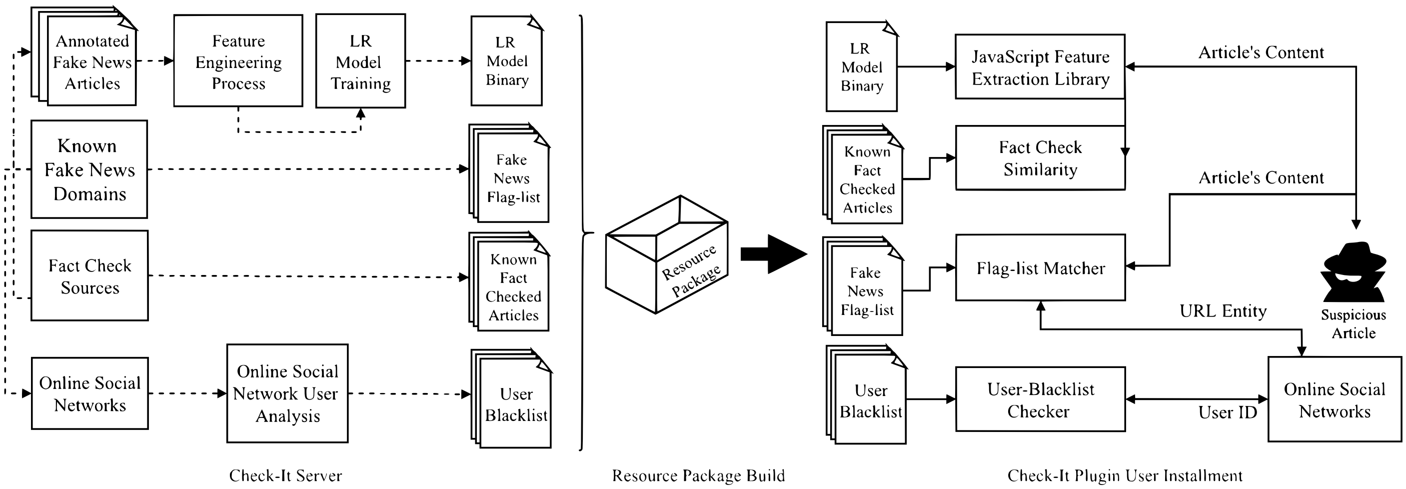


Figure 1: Architectural diagram for the Check-It System.

Analysis is responsible for analyzing user behavior in social networks and produces a User-Blacklist of fake news propagators; and d) **LR Model**, is a classifier trained on linguistic features, which are extracted from fake news datasets using the proposed feature engineering process.

In the following paragraphs, we describe each of the components and explain how the Check-It browser plugin operates at the Check-It Plugin User Installment.

3.1. Fake News Identification

To evaluate the trustworthiness of an article, Check-it passes it through a sequence of steps, each step using a different signal to assess the article's validity. The following signals are integrated into the Check-it pipeline:

Domain Flag-list Signal: Flag-lists refer to well known domains for spreading misinformation (e.g. Kaggle[13], OpenSources[14] and Greek-Hoaxes[15]), annotated and maintained by expert journalists, editors, political and social scientists. The use of flag-lists is considered to be one of the simplest ways

[13] `https://www.kaggle.com/mrisdal/fake-news`
[14] `https://github.com/BigMcLargeHuge/opensources`
[15] `https://github.com/Ellinika-Hoaxes/Greek-Hoaxes-Detector`

for an initial, fast assessment of the trustworthiness of a news article. Although this step does not test the truthfulness of the article itself, it identifies articles originating from sites that engage consistently in disinformation campaigns or propaganda spreading. To this end, Check-It maintains a curated collection of such lists (*Known Fake News Domains*).

Fact Check Similarity Signal: A number of initiatives and organizations, like Politifact[16], Snopes[17], and MediaBiasCheck[18], are dedicated to combating propaganda and hoaxes circulating on the Internet. These sites typically employ professional journalists, political experts, or even people from every side of the political spectrum[19], to do research and comment on the truthfulness of articles [17]. Once the truthfulness or falsehood of an article is established, these websites publicize their findings and associated information (URL, etc.). Check-It capitalizes on fact-checking websites (*Fact Check Sources*) by cross-checking every article processed by its plugin against *Known Fact-Checked Articles* and generating an informative warning when an article happens to be found listed on these web sites.

Online Social Network Signal: Although perpetrators generate false content with the intent to harm, *Online Social Networks* (OSNs) provide the means for spreading it. Recent studies [18] have demonstrated that OSN platforms e.g. Twitter have become mechanisms for massive disinformation campaigns. Since OSNs play an important role in the propagation of fake news, we have incorporated them as another signal in the Check-It toolkit. The idea behind the Check-It OSN signal, similar to Vosoughi et al. [10], is to apply *Online Social Network User Analysis* and provide a dynamic *User-Blacklist*, matching user IDs with a falsity score, indicating the likelihood of a user to post fake news articles. By employing such a list, Check-It is able to warn the users of posts originating from suspicious users. For the purpose of this work, only Twitter is supported due to its massive popularity and the ease-of-access to its data stream via the Twitter Streaming API[20]. In particular, our system consumes tweets from the Twitter stream, identifies

[16]`https://www.politifact.com/`
[17]`https://www.snopes.com/`
[18]`https://mediabiasfactcheck.com/`
[19]`https://www.allsides.com`
[20]`https://developer.twitter.com/en/docs/tweets/filter-realtime/overview`

URLs from Known Fake News Domains, and applies the DeGroot-based user probabilistic model [19] over for the user falsity score calculation, producing as output the User-Blacklist.

Textual Analysis Signal: The signals described so far focus on meta-information retrieved from or associated with the news articles processed by Check-It. The textual analysis signal relies on the actual content of an article (headline and body), leveraging Natural Language (NLP) Processing techniques to extract linguistic features commonly used in fake news [20, 21, 22]. These features are used to train a Machine Learning (ML) based Logistic Regression Model (*LR Model*) over a dataset of *Annotated Fake News Articles*, in order to predict the article's veracity. On the browser, the input features are extracted from the article via the JavaScript Feature Extraction Library that we have implemented.

3.2. Preservation of User's Privacy

Check-It addresses the privacy issue by operating in an overall incognito mode. To do so, Check-It localizes execution by loading the required resources in the browser's local memory. These resources are combined in a Resource Package, which includes the Fake News Flag-lists, the Known Fact Checked Articles, the User-Blacklist, and the binary-produced LR Model. The single client-server communication is taking place during the installation process or the update of the plugin's resources. Specifically, during the installation of the plugin, the Resource Package is built, retrieved, and installed on the user's end. A similar process is applied for any major resource updates.

The localization of the system's execution comes with the trade-off between having proper infrastructure executing the server-side tasks and the heavy computations that stress the user's personal computer with higher memory-footprint and computational workload. Check-It balances this trade-off, by optimizing the local system execution e.g with the use of paralellization. To this end, the following four functional requirements are defined:

- **Preserve User Privacy**: The Check-It plugin should work locally, on the user's web browser, without the need for external communication (i.e. a RESTful APIs), account registration, or HTTP cookies, etc.

- **Highly Confident Identification**: Check-It should label a piece of news as fake if it is highly confident about it (high probability of fake over real).

- **Low Response Time**: All the required resources for the plugin to work, such as the *flag-list* (blacklisted URL domains) and *LR model* (the NLP-based fake news classification model), are efficiently loaded in the user's web browser, and developed so as to have a low response time.

- **Lightweight Computation**: The use of asynchronous processing and parallelization on the users' browsers so as to minimize the load of the plugin.

However, state-of-the-art ML and DL textual models [23, 24, 15, 25, 7] require large amounts and complex features, resulting to thousands of parameters, making them memory-intensive and not appropriate for local execution. To address the above, we develop a *Two-phase Feature Selection Method* (a detailed description is given in Section 4.2) with the intention of reducing the dimensionality of the feature space whilst achieving high classification accuracy. By leveraging a reduced number of features, we train a simpler and more easily interpretable Logistic Regression classifier.

Based on the above, the overall time complexity at the client-side is low. The Domain Flag-list Signal and Online Social Network Signal take $O(1)$ for domain lookup, as they both utilize hashing. The Fact Check Similarity Signal takes $O(p)$ time to execute, where p is the size of fact-checked articles set. The Textual Analysis Signal corresponds to the LR Model and Feature Extraction component respectively. The LR Model takes $O((f+1)c)$ time, with f being the number of features used, and c the number of classes. In our case, we have a binary classification, thus $c = 1$ and the final time is $O(f+1)$. The Feature Extraction component is mostly comprised of trivial tasks e.g. lookups, with $O(w)$ and w being the size of words in the article. Only the Part-of-Speech tagging is considered to be a bottleneck, with a time of $O(slt+t)$ with s being the size of article sentences, l the average sentence word size, and t the total number of tags.

4. Feature Engineering Process

In this section, we describe the Check-It feature engineering process, which comprises two equally important sub-processes. The first one is the feature extraction which processes the article texts and generates a series of textual features; and the second is the two-phase feature selection approach. The final set of features are then given as input to the LR Model.

4.1. Feature Extraction: Stylistic, Complexity and Psychological

Fake news detection in traditional news media mainly relies on news content, such as the headline and the body of an article. At the Check-It Plugin User Installment, the system computes different linguistic features from these article sections and feeds them to the LR Model for classification (Figure 1). We group these features into three broad categories: *Stylistic*, *Complexity* and *Psychological*. More details for the extracted features are included in Section 9.

Examples of the Features Extracted		
Stylistic	**Complexity**	**Psychological**
# of "I" pronouns	Gunning_fog	# of analytical words
# of all capital letters	SMOG Grade	# of negations
# of stop words	Flesh-Kincaid	# of slang words
# of Verbs	Yules_k	# of power words
# of quotes (")	Coleman Liau	# of casual words
# of adverbs	Dale Chall	# of emotion words
# of "We" pronouns	Brunets W	# of risk words
# of full stops (.)	Honores R	# of certainty words
# of words	# of happax legomena	# of power words
# of lines	# of happax dislegomena	# of affiliation words

Table 1: A sample of the extracted features divided into the three categories. The symbol '#' refers to the frequency of the respective feature.

Stylistic Features: represent the syntax and writing style of the article. They are calculated based on widely known NLP techniques. Text style features include the frequency of stop-words, punctuation, quotes, negations, and words that appear in all capital letters, whereas syntactical features include the frequency of Part-of-Speech tags in the text.

Complexity Features: capture the overall intricacy of an article or headline. This intricacy can be computed based on several word-level metrics that include readability indices and vocabulary richness. Specifically, we compute the Gunning Fog, SMOG Grade, and Flesh-Kincaid grade level readability indices. Each measure computes a grade-level reading score based on the number of complex words (e.g. over 3 syllables). A higher index means a document takes a higher education level to read. Moreover, we compute the Type-Token Ratio, which can be defined as the number of unique words divided by the total number of words in the article. In order to capture the vocabulary richness of the content, we also compute the number of hapax legomenon and dis legomenon which correspond to phrases that occur only once and twice within a context.

Psychological Features: are based on the count of words found in expert dictionaries that are associated with different psychological processes. These dictionaries include the negative and positive opinion lexicon [26], and the moral foundation dictionary [27]. The sentiment score is computed via the AFINN sentiment lexicon [28], a list of English terms manually rated for valence. The AFINN sentiment score is defined as an integer number between -5 and +5, indicating the negative and positive scores respectively.

4.2. Feature Selection: The Two-phase Method

The extensive amount of features (535 features) result in a high dimensional space, while, in our previous work [6], it produced a DL model with an extensive number of parameters, making it incompatible with a prevalent web browser due to memory limitations. Therefore, the application of a feature selection method is imperative. For this purpose we applied a custom two-phase feature selection technique, combining an embedded method (L2-Regularization) and a wrapper method (Genetic Algorithm). The proposed method consists of the following two steps:

1. The L2-regularization feature selection method [29] applied to the raw set of extracted features. This method produces a ranking of the input features according to their importance (described in detail in Section 4.2.1).
2. The intermediate ranked features are given as input to the Genetic Algorithm (GA) through an iterative process, which produces the subset of optimal features to be used. This step is described in more details in Section 4.2.2.

4.2.1. L2-Regularization Feature Selection Method

L2-regularization method is considered an embedded method as it performs feature weighting based on regularization models [29]. The weighting is applied using an objective function that minimizes the fitting errors and minimizes the feature coefficients. Regularization consists of attaching a penalty to the feature coefficients of any linear machine learning model to increase the generalization of the model, reduce multicollinearity, and avoid overfitting. In the linear model regularization, the penalty parameter (λ) is applied over the coefficients (β) that multiply each of the predictors (p). L2-Regularization uses the ridge regression model $\lambda \sum_{j=1}^{p} \beta_j^2$ which encourages the sum of the squares of the parameters to be small.

In our approach, the L2 penalty (λ) is applied to the LR Model to maximize a penalized version of the cost function (1). Combining the penalty term $\lambda \sum_{j=1}^{p} \beta_j^2$ and the cost function of LR (1), we conclude to equation (2). The ridge regression model instead of eliminating features, it ranks the feature coefficients in ascending order based on their absolute value [30].

$$\sum_{i=1}^{N} \left\{ y_i \beta^T x_i - log(1 + e^{\beta^T x_i}) \right\} \tag{1}$$

$$max_\beta \left\{ \sum_{i=1}^{N} y_i \beta^T x_i - log(1 + e^{\beta^T x_i}) - \lambda \sum_{j=1}^{p} \beta_j^2 \right\} \tag{2}$$

4.2.2. Genetic Algorithm as a Feature Selection Method

The GA is an optimization problem-solving method proposed by [31]. With respect to the feature selection problem, each solution in the population of genotypes represents a candidate solution for selecting a feature subset. Each gene represents a feature, so the length of the genotype is equal to the total number of input features available. The classification performance of the LR classifier is used as the *fitness function* (objective evaluation function) which determines the likelihood of the genotype to survive on the next iteration. The ones with the highest fitness value survive in the next generation and two of them (*parents*) are randomly selected to produce an *offspring* using the crossover or mutation processes on each iteration.

For this work, as a fitness function, we use the logarithmic classification loss and the objective goal is to minimize the classification error (*log-loss*). As a termination criterion, we use the number of iterations to be equal to 20. We apply a uniform crossover and a one-point mutation. The population size at each generation is equal to *generation-size = 500* and the number of best individuals (genomes) to survive to next generation is equal to *generation-best-ratio = 20*. The parameters have been selected such as to reduce the probability of overfitting by generating a large number of new solutions at each generation (*generation-size - generation-best-ratio = 480*). We choose a small number of iterations to reduce the execution time of our approach.

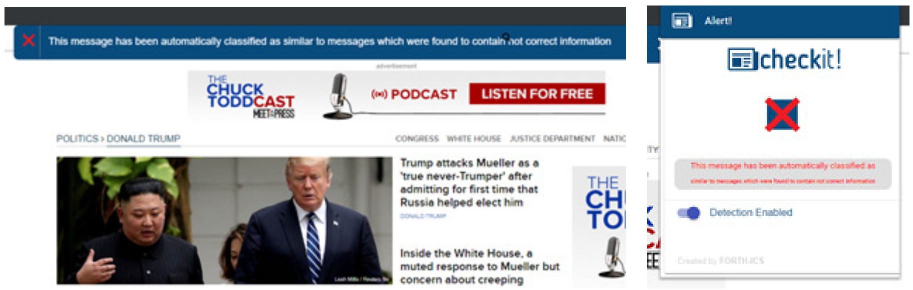
To maximize the performance of the proposed two-phase feature selection process, we iteratively provide the ranked features of the L2-regularization output as input to the GA, through a brute force search. In each iteration, we feed the GA with the top-k features, where k increases at each iteration and it ranges between 10<k<total size of features.

5. Check-It Plugin

In this section, we introduce the Check-It browser plugin which readily available in the browsers' marketplace[21]. The aforementioned components are incorporated in the plugin which is compatible with Google's Chrome and Mozilla's Firefox web browsers. While the users surf the web and read news articles online, Check-It runs in the background, analyzes locally the articles that a user reads, and provides a warning when an article has suspicious content based on credibility.

When a user first loads an article's URL, the *Flag-list Matcher* (described in section 3.1) is applied, isolating the article's source domain from the URL and checking it against existing *Known Fake News Domains*. As a result, Check-it warns the user with an exclamation mark and a popup (depicted in Figure 5), indicating the reason for suspicion with an appropriate message such as: "This domain appears as questionable in the list provided by ...". If the domain is not present in any of the flag-lists, then Check-it applies a similarity check via the Fact Check Similarity component (described in section 3.1). If the article is closely similar to other *Known Fact Checked Articles*, the user

[21] `http://bit.ly/2pRBGqC`



(a) Check-It Message (b) Check-It Popup

Figure 2: Screenshots from the Check-It warning message and popup for an article that is automatically classified as suspicious content.

receives a warning message like: "This article appears similar to: ...". As a final step, Check-it provides a content suspicion analysis through the *LR Model* (described in section 3.1). Using the *Javascript Feature Extraction Library*, Check-It extracts the appropriate features, and checks the suspicion of the article's content by predicting its veracity based on a probability score. If the score passes a pre-defined threshold, the user receives the warning message of "The content of this article is classified as very suspicious". In our case, we set the threshold to 0.90. The above pipeline is also applied to URLs and articles shared within online social networks (OSNs). By using the *User-Blacklist Checker*, the user is also get informed when he looks at posts in OSNs by users from the *User Blacklist*.

To summarize, Check-it examines both the source and content of the article and if the article is flagged as suspicious, it warns the user with the analogous explanatory message. By repeatedly using Check-It, the user eventually adopts a pattern for verifying the content of an article before sharing it online.

6. Evaluation

In this section, we evaluate the proposed feature engineering approach and the overall performance of the Check-It. We also describe our findings of a pilot use-case, regarding the UI-UX of the plugin based on a user questionnaire.

All the performance experiments take place on a Virtual Machine with Ubuntu 16.4, 16VCPUs, and 32GB of RAM.

6.1. Dataset Overview

Several datasets on fake news and factual statements are publicly available online i.e. LIAR [32], CREDBANK [33], Fake News Corpus and, BS Detector. For the evaluation of the proposed methodology, we utilize two comprehensive fake news datasets [22] collected by social media, and both classify their articles from expert journalists: the PolitiFact and BuzzFeed[23].

BuzzFeed News: This dataset comprises a complete sample of news that was published on Facebook, originating from 9 news outlets over the period of a week during the 2016 U.S. elections. Each Facebook post is attached with a news article that was fact-checked by 5 BuzzFeed journalists. In [23], the initial dataset is further enriched by adding the linked articles, attached media, and relevant metadata. In this work, we use the older version which consists of 182 news articles.

PolitiFact: This dataset consists of a list of fake news articles and their corresponding news content that were scraped from their respective websites. PolitiFact is a fact-checking organization, employing journalists to validate the factual veracity of news and other online content. A set of 240 articles labeled by PolitiFact journalists as fake or real were collected, along with a scraped version of the analogous news articles.

6.2. Feature Selection Evaluation

To evaluate the feature engineering process of Check-It, first, we compare our two-phase feature selection approach (L2-GA) with other well-known feature selection methods that belong to different categories: filter-based, wrappers, embedded, and hybrid methods. More specifically, we compare it against: i) the following filter-based methods: F-test, Mutual Information (MI) [34], and Chi-Square (χ^2) test [35], ii) the following wrapper methods: Sequential Backward Floating Selection (SBFS) [36], Sequential Forward Floating Selection (SFFS) [36], the standalone GA method [34] and iii) the following embedded approaches: the L1 and L2 regularization methods [29]. Regarding the comparison with the hybrid methods, we compare it with a hybrid

[22] `https://github.com/KaiDMML/FakeNewsNet/tree/old-version`

[23] The specific datasets were chosen to be able to compare with [23, 7]

method integrated into the Autofeat Python library, [37][24]. Additionally, since L2 produces a sorted list of all the features, from the most to the least important (rank-based method), for a more fair comparison with our hybrid approach (L2 - GA), we also combine GA method with the rank-based methods: F-test, MI, Chi-Square. For all the rank-based methods (F-test, MI, Chi-Square, L2) and the hybrid methods (F-test - GA, MI-GA, Chi-square - GA, L2-GA), we apply a brute force analysis to identify the number of the top-ranked features that maximize the classification performance. We also used the t-test to assess the statistical significance of our results. All the implemented classifiers for both datasets are evaluated using 5-fold cross-validation.

6.2.1. Results

Considering the application of a feature selection method before the implementation of the classification task of fake news detection using the LR model, as Tables 2 and 3 depict, almost all of the feature selection methods (except Chi-square applied on BuzzFeed), improve significantly the classification performance. The difference in the model performance after applying L2-GA comparing to the model performance without applying any feature selection method is statistically significant ($p < 0.05$). Moreover, all of the feature selection methods decrease the number of features, which is essential, especially in a plug-in application where the computational time and memory-usage must be limited. Specifically, the proposed two-phase method reduces the features by 86% on the PolitiFact dataset (74 features) and by 80% for the BuzzFeed dataset (106 features).

Our approach (L2-GA) outperforms the rest of the feature selection methods for both datasets and for most of the methods, the difference is statistically significant ($p < 0.05$) as also shown in Table 3. However, by taking into consideration both the reduced number of features and the model performance (f1 score), L2-GA performance outperforms the majority of the well-known, standalone and hybrid feature selection methods, in both datasets.

Selected Features: The selected features show that there is a significant difference in the content and titles of fake and real news articles. Many of our findings are in line with those of other works in the literature, such as the

[24] `https://pypi.org/project/autofeat/`

Category	Method	Dataset									
		Politifact					**Buzzfeed**				
		#	**Acc.**	**Prec.**	**Rec.**	**F1**	#	**Acc.**	**Prec.**	**Rec.**	**F1**
Without Feature Selection		535	0.706	0.708	0.706	0.705	535	0.725	0.735	0.725	0.722
Filters	F-Test	9	0.706	0.709	0.706	0.705	75	0.769	0.782	0.769	0.767
	MI	143	0.706	0.712	0.706	0.704	2	0.771	0.773	0.771	0.770
	χ^2	28	0.710	0.715	0.710	0.709	80	0.694	0.701	0.694	0.691
Wrappers	SFFS	195	0.861	0.865	0.861	0.860	271	0.907	0.910	0.907	0.907
	SBFS	67	0.840	0.843	0.840	0.840	38	0.907	0.908	0.907	0.907
	GA	234	0.798	0.802	0.798	0.798	247	0.841	0.847	0.841	0.840
	L1	29	0.790	0.792	0.790	0.790	16	0.781	0.793	0.781	0.779
	L2	71	0.827	0.831	0.827	0.826	37	0.879	0.883	0.880	0.879
Combination	Autofeat	14	0.769	0.771	0.769	0.768	7	0.775	0.793	0.775	0.771
	F-Test - GA	66	0.866	0.870	0.866	0.865	101	0.901	0.904	0.901	0.901
	MI - GA	357	0.870	0.874	0.870	0.870	491	0.880	0.885	0.880	0.880
	χ^2- GA	68	0.857	0.861	0.857	0.857	152	0.895	0.901	0.895	0.895
	L2 - GA	**74**	**0.907**	**0.910**	**0.907**	**0.907**	**106**	**0.946**	**0.947**	**0.946**	**0.946**

Table 2: Comparison of accuracy, precision, recall and f1 score for the proposed two-phase feature selection method with other widely known feature selection methods applied on the PolitiFact and Buzzfeed datasets. The '#' column represents the final number of features per method. The best results are marked in bold. (*Acc.* stands for Accuracy, *Pre.* for Precision, *Rec.* for Recall and *F1* for F1 score)

length of real news being greater than fake news [8, 16, 25]. Consistently, throughout the articles, we find that fake news articles use the plural pronoun *"we"* significantly more than real news articles. Our interpretation is that the authors of fake news articles try to emotionally invoke their readers to believe their stories by presenting that all we share the same concerns [38, 2, 23]. Also, fake news articles seem to use significantly more litigious words for the deliverance of justice and law and order than the real news articles. This justifies the literature findings that fake news is related to the rise of political polarization [39]. The title has also shown significant differences between fake and real. We found that fake news titles contain more words in all capital letters, with more proper nouns and negative sentiment. In contrast with the titles of fake news, real news titles contain more nouns and stopwords [8].

6.3. Check-It Performance Evaluation

In this section, we compare the performance of the LR model, trained on features extracted using the proposed feature engineering process, against

Category	Method	Dataset			
		Politifact		Buzzfeed	
		F-score	**p-value**	**F-score**	**p-value**
No Feature Selection Applied		0.705 *** (± 0.020)	1.33E-05	0.722 *** (±0.054)	8.00E-05
Filters	F-Test	0.705 *** (±0.073)	0.001	0.767 *** (±0.057)	0.001
	MI	0.704 *** (±0.031)	3.30E-05	0.770 *** (±0.089)	0.006
	χ^2	0.709 *** (±0.021)	2.00E-05	0.691 *** (±0.082)	3.60E-04
Wrappers	SFFS	0.860 (±0.113)	**0.457**	0.907 (±0.059)	**0.270**
	SBFS	0.8397 * (±0.047)	0.055	0.907 (±0.056)	**0.255**
	GA	0.798 ** (±0.060)	0.015	0.840 ** (±0.076)	0.032
	L1	0.790 ** (±0.075)	0.023	0.779 ** (±0.024)	1.93E-05
	L2	0.826 (±0.104)	**0.182**	0.879 (±0.042)	0.029
Combination	Autofeat	0.768 *** (±0.049)	0.002	0.772 *** (±0.032)	3.78E-05
	F-Test - GA	0.865 (±0.039)	**0.161**	0.901 *** (±0.037)	0.094
	MI - GA	0.870 (±0.019)	**0.116**	0.879 ** (±0.036)	0.020
	χ^2- GA	0.857 (±0.046)	**0.133**	0.895 * (±0.122)	0.052
	L2 - GA	**0.907 (±0.038)**	-	**0.946 (±0.028)**	-

Table 3: Comparison of f-score and p-value of the t-test. The standard deviation is displayed between parenthesis. T-Test p-values: *** $p < 0.01$, ** $p < 0.05$, and * $p < 0.1$. The best f1 scores and the statistically insignificant different results are marked in bold.

the Deep Neural Network model (DNN) introduced in the initial version of Check-It [6]. In addition, we compare the regression model performance ($L2{-}GA_{LR}$) with several state-of-the-art methods on fake news detection [23, 7, 15]. Note that for a fair comparison we chose baselines that only consider news contents, similar to our approach, and also replicate their training configurations. For the comparisons, we use the outcomes reported in the aforementioned publications [23, 15, 7].

Shu et al. [23] applies multiple classifiers on the PolitiFact dataset using one-hot vector representation for each news article. The classifiers used in that work include a Support Vector Machine (SVM), a Logistic Regression (LR), a Naive Bayes (NB). Additionally, the authors include a deep learning approach, namely a Convolutional Neural Network (CNN), trained over word embeddings of the articles. Shu et al. [15] train an SVM classifier using the vectorized output from LIWC lexicon[40]. Potthast et al. [7] introduced four different Random Forest (RF) classifiers. The features of the four classifiers were extracted from the style and topic of the news content (NC). Such features include character n-grams, stop words, part-of-speech, as well as word frequencies and several readability indices. Two of the aforementioned classifiers consider the political orientation of the articles, ORF_{STYLE} and ORF_{TOPIC}, whereas the other two are generic, namely GRF_{STYLE} and GRF_{TOPIC}.

In addition to the utilization of the news' content, recent approaches extend the task with multiple characteristics including user and publisher information, as well as, their relation with the content to identify the veracity of an article. Based on this, we also compare the results of applying our approach on both datasets with the results published in [15] where the authors proposed the TriFN framework which consolidates publisher-news relations and user-news interactions simultaneously.

For a fair comparison, we replicated the evaluation configuration of the different approaches. For the comparison with the works of Shu et al. [23] and Shu et al. [15], we split the data into 80% training and 20% testing and presented the averages over 10 iterations. For the comparison with the work of Potthast et al. [7], we applied a 3-cross validation and presented the average scores.

6.3.1. Results

Compared to the recently published state-of-the-art works, our approach is superior in detecting fake news based only on textual-context, by utilizing the articles' title and body, as displayed in Table 5. In addition, Table 4 shows that the integrated two-phase feature selection process that uses L2 regularization and Genetic Algorithm (GA) outperforms the Deep Neural Network model (DNN) which introduced in the preliminary version of Check-It [6].

Model	Dataset	Acc.	Pre.	Rec.	F1
$Check-It_{DNN}$	PolitiFact	0.728	0.734	0.727	0.725
	BuzzFeed	0.715	0.719	0.715	0.714
$L2-GA_{LR}$	PolitiFact	**0.907**	**0.905**	**0.915**	**0.908**
	BuzzFeed	**0.946**	**0.937**	**0.957**	**0.946**

Table 4: Comparison of the Check-It($L2-GA_{LR}$) with the initial version ($Check-It_{DNN}$). The best results are marked in bold.

Regarding the TriFN framework [15], the difference in the performance can be considered negligible, because the f1-score of our approach, is just 2.8%, and 7.6% higher than TriFN on PolitiFact and Buzzfeed dataset respectively. However, our approach differs from the TriFN framework in the fact that it considers only the extraction of content linguistic features. TriFN takes into consideration metadata that includes information regarding the publisher and user interactions on online social media. Even though the combination of all these metadata captures significant knowledge regarding fake and real news, our approach manages to outperform TriFN by considering only the textual information.

For the rest of the comparisons, we have a significant difference in the classification performance, even with complex DNNs, which as the experiments define, they have downsides, especially when the training happens on high dimensional data with few entries in the datasets.

Reference	**Dataset**	**Input**	**Acc.**	**Pre.**	**Rec.**	**F1**
Shu et al. [23] (2018)	Politifact	NC_{SVM}	0.580	0.611	0.717	0.659
		NC_{LR}	0.642	0.757	0.543	0.633
		NC_{NB}	0.617	0.674	0.630	0.651
		NC_{CNN}	0.629	0.807	0.456	0.583
Potthast et al. [7] (2017)	BuzzFeed	$STYLE_{GRF}$	0.550	0.520	0.525	0.520
		$TOPIC_{GRF}$	0.520	0.515	0.515	0.510
		$STYLE_{ORF}$	0.550	0.535	0.540	0.535
		$TOPIC_{ORF}$	0.580	0.555	0.555	0.560
Shu et al. [15] (2019)	PolitiFact	$LIWC$	0.688	0.725	0.617	0.666
		$TriFN$	0.878	0.867	0.893	0.880
	BuzzFeed	$LIWC$	0.719	0.722	0.732	0.709
		$TriFN$	0.864	0.849	0.893	0.870
Our Approach	PolitiFact	$L2-GA_{LR}$ †	**0.903**	**0.905**	**0.907**	**0.903**
		$L2-GA_{LR}$‡	**0.875**	**0.877**	**0.875**	**0.874**
	BuzzFeed	$L2-GA_{LR}$ †	**0.924**	**0.927**	**0.924**	**0.924**
		$L2-GA_{LR}$ ‡	**0.899**	**0.904**	**0.899**	**0.899**

Table 5: Overall results on the comparison of our feature engineering approach with the state-of-the-art works on fake news detection using both datasets. The best results are marked in bold. (*Acc.* stands for Accuracy, *Pre.* for Precision, *Rec.* for Recall and *F1* for F1 score). †refers to Shu [15] evalutation and ‡refers to Potthast [7] evaluation.

6.4. Check-It Plugin UI-UX Evaluation

Despite the satisfactory performance of Check-It's individual components, we additionally evaluated its overall performance in a controlled environment with respect to fake news identification accuracy and system's usability. Thus, we created a survey that also served as a test-case to receive users' feedback from a pilot use.

At the phase of the pilot use, 17 users (undergraduate, postgraduate students, and faculty members with different academic/professional backgrounds) participated in the evaluation of the Check-it plug-in. Specifically, the participants consisted of 5 undergraduate students of Computer Science (CS) major, 3 postgraduate students of CS major, 2 post-doctoral fellows with a background in Social Sciences, 2 faculty members of CS background, 4 undergraduate students of Journalism major and 1 graduate, and experienced journalist.

The test case prompted the participants to utilize their critical thinking in order to investigate the veracity of certain news titles, both true and fake, that receive a lot of attention online. The participants were asked to use the Check-It plugin for their assessments.

6.4.1. Results

The results of the evaluation indicate that 91% of the participants made correct decisions during their credibility assessments. For the veracity rating, we made use of the Politifact Likert scale ratings, with the available options of: *True, Mostly True, Half True, Mostly False* and *False*. Most of the correct submissions were labeled as True (32.1%), Mostly True (25%), and False (21.4%). Regarding the incorrect submissions, all of them were labeled as Mostly False (100%).

The credibility forms were followed by a series of 13 questions regarding the accuracy and usability of the plugin. The key findings of these questions are the following:

Usefulness: All of the participants stated that the plugin was either very (50%), quite (30%) or simply useful (20%), with informative messages regarding the reasons for its annotations (57% informative, 29% quite informative and 14% very informative).

Accurate: All of the participants stated that the plugin was either very (57%) or quite (43%) accurate. When asked if they observed any miss-classifications of authoritative articles as fake, 20% answered positively. Finally when asked if Check-It is able to achieve its purpose for assessing the support of the detection of fake news, 57% answered that they strongly agree and 43% agree.

Recommendations: All of the participants stated that they would use the Check-It plugin during their daily life or their workspace. Most of them would also recommend the use of the plugin to other people (71.4% strongly recommend and 14.3% recommend).

Aspect Importance: All of the aspects provided by the plugin were marked equally important, with the most important feature being the GDPR compliance (25%). Focusing on the GDPR, 43% of the participants stated that they would not use the plugin if it was not GDPR compliant.

7. Discussion

In the previous section, we provided a comparative analysis of the Check-It feature engineering process with well-known feature selection methods along with the comparison with state-of-the-art fake news detection approaches by conducting experiments on two real-world datasets.

Based on the results, Check-it significantly outperforms the four state-of-the-art fake news detection methods by at least 3.33% in F1-score, and by achieving accuracy over 90%. Considering the extensive comparison, we undoubtedly prove the importance of our approach as a high-quality feature engineering process and the Check-It as a promising plugin to efficiently automatically detect fake news using linguistic features.

Moreover, according to the results of evaluating its overall performance, Check-It is a tool that contributes to increasing the use of critical thinking towards identifying fake news, and at the same time, it respects the users' privacy. To justify even more the effectiveness of Check-It as a web-browser tool, we also provide an extensive comparison of the Check-It plugin with of six available fake news detection internet browser plugins. Our comparison is contacted based on eight functionality features. The first three characteristics are based on the signals used as input by the plugins, namely the use of domain blacklists, the article content-based (body and headline)

Extension	Black-listed domains	Server-Site API	Content Analysis	Network Analysis	Similarity Check	Account Required	Feedback	Free
FirstDraft	✗	✗	✗	✗	✗	✓	✓	✓
B.S. Detector	✓	✗	✗	✗	✗	✗	✓	✓
NewsGuard	✓	✓	✗	✗	✗	✓	✓	✗
TrustedNews	✗	✓	✓	✗	✗	✗	✓	✓
FakerFact	✗	✓	✓	✗	✗	✗	✓	✓
TweetCred	✗	✓	✗	✓	✗	✓	✗	✓
Check-It	✓	✗	✓	✓	✓	✗	✗	✓

Table 6: Fake news detection plugin comparison of provided functionalities.

analysis, and propagation and social network analysis. The plugin's utilization of client-server communication via an API, account management, and the overall user privacy indicator is another feature we deem important. An additional feature is the use of similarity checks between articles and black-listed domains, or annotated articles from experts (i.e., fact-checking organizations). Finally, free-of-charge and user registration (account sign-up) are also included in the comparison functionalities. All of the comparisons are summarized in Table 6.

Plugins that are based on domain blacklist are B.S. Detector and NewsGuard. These plugins prove the simplicity and potency of a single curated list of untrusted domains. Their difference is that NewsGuard incorporates the corrections and clarifications of these domains on questionable articles, as well as the distinction between objective and subjective articles. Similar to NewsGuard, plugins such as TrustedNews and FakerFact, focus solely on the article-level and leverage the power of Machine Learning (ML) models to analyze the content of an article to give a signal for its credibility. Specifically, TrustedNews examines the objectivity of a piece of news, on a sentence level, and produces an overall score to help the user decide whether it's trustful or not. FakerFact analyzes the intent of the article and using its own AI (named Walt) and informs the users for the article's purpose, e.g., satire, bias, sensational, etc. Following the same philosophy with TrustedNews, FakerFact gives some indications and lets the decision of the article's veracity to the user.

Despite the good results from analyzing the text, the examination of the medium where the news is published can also be an efficient way in the identification of misinformation, as we show in Section 2.2. TweetCred[5], declares the credibility of a tweet based on information related to it, including content, author, retweets, URLs, and other metadata.

However, the main drawback of the previously mentioned plugins is the utilization of a single signal of information. Specifically, B.S Detector and NewsGuard utilize only blacklists, TrustedNews and FakerFact use only content analysis, and TweetCred focuses only on network characteristics. Due to the complexity of fake news detection, a single signal does not always capture all the available knowledge to produce accurate results. To the best of our knowledge, Check-It is the only plugin that combines blacklists, content and network analysis, and also similarity check. The combination of these signals provides a deeper understanding of the news's credibility.

Moreover, most of the works (i.e., NewsGuard, TrustedNews, FakerFact, TweetCred) employ server-side APIs with constant communication to annotate the news, and monitor user reading habits, utilize HTTP cookies, request permissions such as access to the user's internet browsing history (TrustedNews), and require, account registration (FirstDraft, NewsGuard, and TweetCred). Of course, these actions are to have better results and higher accuracy in identifying fake news articles, but may also have a negative impact and making users reluctant in using them during their daily browsing routine.

Also, TweetCred and Check-It have yet to incorporate the ability for users to report untrusted articles and provide feedback. Check-It will have such functionality in the following releases. Lastly, it's worth mentioning that, except NewsGuard, all of the aforementioned approaches are available without any fees.

To sum up, the results of the comparison with the other plugins show that Check-It is the only plugin that combines multiple information signals and it respects users' privacy.

8. Conclusion

In this paper, we presented Check-It, a fake news detection system, developed as a web browser plugin, and a feature engineering approach, which acts as an essential part of the system. Check-It manages to address the defined challenges, by effectively combining a set of diverse signals as a form of the pipeline, to accurately classify fake news articles and timely inform the user, whilst securing user's privacy and smooth experience.

Through the extensive evaluation, the potential of our system, as well as the overall performance in timely and effectively identifying false news is presented. The current work serves as an extension of the initial Check-It work [6], presenting the feature selection method, which produces results that outperform our previous work, and state-of-the-art, with the use of simple ML models such as LR.

As a future work, we are planning to expand our feature space using hyper-partisanship and bias indications via framing [41], as well as to provide a more thorough investigation on the resource utilization and optimization on the client-side.

In conclusion, Check-it aims to take a bold step towards detecting and reducing the spread of misinformation on the Web. To do so, it empowers its users with the tools they need to identify fake news. The novelty of Check-It is the combination of a variety of signals, incorporated in a pipeline, ranging from domain name flag-lists to deep learning approaches.

Acknowledgement

This work is partially supported by the Google Digital News Initiative grant (Check-it) and by the Cyprus Research and Innovation Foundation through COMPLEMENTARY/0916/0036 project.

References

[1] S. Aral, D. Eckles, Protecting elections from social media manipulation, Science 365 (6456) (2019) 858–861. doi:10.1126/science.aaw8243. URL https://science.sciencemag.org/content/365/6456/858

[2] H. Allcott, M. Gentzkow, Social Media and Fake News in the 2016 Election, Journal of Economic Perspectives 31 (2) (2017) 211–236. doi:10.1257/jep.31.2.211. URL http://pubs.aeaweb.org/doi/10.1257/jep.31.2.211

[3] S. Aral, The Hype Machine: How Social Media Disrupts Our Elections, Our Economy, and Our Health and How We Must Adapt, Currency, 2020.

[4] P. N. Howard, Lie Machines: How to Save Democracy from Troll Armies, Deceitful Robots, Junk News Operations, and Political Operatives, Yale University Press, 2020.

[5] A. Gupta, P. Kumaraguru, C. Castillo, P. Meier, Tweetcred: A real-time web-based system for assessing credibility of content on twitter, CoRR abs/1405.5490 (2014). arXiv:1405.5490.
URL http://arxiv.org/abs/1405.5490

[6] D. Paschalides, C. Christodoulou, R. Andreou, G. Pallis, M. D. Dikaiakos, A. Kornilakis, E. Markatos, Check-it: A plugin for detecting and reducing the spread of fake news and misinformation on the web, in: Proceedings - 2019 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2019, Association for Computing Machinery, Inc, 2019, pp. 298–302. arXiv:1905.04260, doi:10.1145/3350546.3352534.

[7] M. Potthast, J. Kiesel, K. Reinartz, J. Bevendorff, B. Stein, A stylometric inquiry into hyperpartisan and fake news, CoRR abs/1702.05638 (2017). arXiv:1702.05638.
URL http://arxiv.org/abs/1702.05638

[8] B. D. Horne, S. Adali, This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news., CoRR abs/1703.09398 (2017).
URL http://dblp.uni-trier.de/db/journals/corr/corr1703.html#HorneA17

[9] L. Wang, Y. Wang, G. De Melo, G. Weikum, Five shades of untruth: Finer-grained classification of fake news, Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2018 (2018) 593–594doi:10.1109/ASONAM.2018.8508256.

[10] S. Vosoughi, D. Roy, S. Aral, The spread of true and false news online, Science 359 (6380) (2018) 1146–1151. arXiv:https://science.sciencemag.org/content/359/6380/1146.full.pdf, doi:10.1126/

science.aap9559.
URL https://science.sciencemag.org/content/359/6380/1146

[11] C. Castillo, M. Mendoza, B. Poblete, Information credibility on twitter, in: Proceedings of the 20th International Conference on World Wide Web, WWW '11, Association for Computing Machinery, New York, NY, USA, 2011, p. 675–684. doi:10.1145/1963405.1963500.
URL https://doi.org/10.1145/1963405.1963500

[12] Z. Jin, J. Cao, Y. Zhang, J. Luo, News verification by exploiting conflicting social viewpoints in microblogs, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI'16, AAAI Press, 2016, p. 2972–2978.

[13] X. Dong, U. Victor, S. Chowdhury, L. Qian, Deep two-path semi-supervised learning for fake news detection, CoRR abs/1906.05659 (2019). arXiv:1906.05659.
URL http://arxiv.org/abs/1906.05659

[14] N. Ruchansky, S. Seo, Y. Liu, CSI: A Hybrid Deep Model for Fake News Detection, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM '17, ACM, New York, NY, USA, 2017, pp. 797–806. doi:10.1145/3132847.3132877.
URL http://doi.acm.org/10.1145/3132847.3132877

[15] K. Shu, S. Wang, H. Liu, Beyond news contents: The role of social context for fake news detection, WSDM 2019 - Proceedings of the 12th ACM International Conference on Web Search and Data Mining (December) (2019) 312–320. arXiv:1712.07709, doi:10.1145/3289600.3290994.

[16] M. L. Della Vedova, E. Tacchini, S. Moret, G. Ballarin, M. DiPierro, L. de Alfaro, Automatic online fake news detection combining content and social signals, in: 2018 22nd Conference of Open Innovations Association (FRUCT), 2018, pp. 272–279.

[17] J. Thorne, A. Vlachos, Automated fact checking: Task formulations, methods and future directions, in: Proceedings of the 27th International Conference on Computational Linguistics, Association for Computational Linguistics, Santa Fe, New Mexico, USA, 2018, pp. 3346–3359.

[18] D. M. J. Lazer, M. A. Baum, Y. Benkler, A. J. Berinsky, K. M. Greenhill, F. Menczer, M. J. Metzger, B. Nyhan, G. Pennycook, D. Rothschild, M. Schudson, S. A. Sloman, C. R. Sunstein, E. A. Thorson, D. J. Watts, J. L. Zittrain, The science of fake news, Science 359 (6380) (2018) 1094–1096. doi:10.1126/science.aao2998.
URL https://science.sciencemag.org/content/359/6380/1094

[19] M. H. DeGroot, Reaching a consensus, Journal of the American Statistical Association 69 (1974) 118–121. doi:10.2307/2285509.

[20] B. D. Horne, S. Adali, This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news, CoRR abs/1703.09398 (2017). arXiv:1703.09398.
URL http://arxiv.org/abs/1703.09398

[21] V. Rubin, N. Conroy, Y. Chen, S. Cornwell, Fake news or truth? using satirical cues to detect potentially misleading news, in: Proceedings of the Second Workshop on Computational Approaches to Deception Detection, Association for Computational Linguistics, San Diego, California, 2016, pp. 7–17. doi:10.18653/v1/W16-0802.
URL https://www.aclweb.org/anthology/W16-0802

[22] Y. Wang, G. de Melo, G. Weikum, Five shades of untruth: Finer-grained classification of fake news, 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM) (2018) 593–594.

[23] K. Shu, S. Wang, H. Liu, Exploiting tri-relationship for fake news detection, CoRR abs/1712.07709 (2017).

[24] K. Shu, D. Mahudeswaran, S. Wang, D. Lee, H. Liu, Fakenewsnet: A data repository with news content, social context and dynamic information for studying fake news on social media, CoRR abs/1809.01286 (2018). arXiv:1809.01286.
URL http://arxiv.org/abs/1809.01286

[25] K. Shu, D. Mahudeswaran, H. Liu, FakeNewsTracker: a tool for fake news collection, detection, and visualization, Computational and Mathematical Organization Theory 25 (1) (2019) 60–71. doi:10.1007/s10588-018-09280-3.

[26] B. Liu, M. Hu, J. Cheng, Opinion observer: Analyzing and comparing opinions on the web, in: Proceedings of the 14th International Conference on World Wide Web, WWW '05, ACM, New York, NY, USA, 2005, pp. 342–351. doi:10.1145/1060745.1060797.
URL http://doi.acm.org/10.1145/1060745.1060797

[27] J. Graham, J. Haidt, B. Nosek, Liberals and conservatives rely on different sets of moral foundations, Journal of personality and social psychology 96 (2009) 1029–46. doi:10.1037/a0015141.

[28] F. Å. Nielsen, A new evaluation of a word list for sentiment analysis in microblogs, CoRR abs/1103.2903 (2011). arXiv:1103.2903.
URL http://arxiv.org/abs/1103.2903

[29] A. Y. Ng, Feature selection, L1 vs. L2 regularization, and rotational invariance, in: Proceedings, Twenty-First International Conference on Machine Learning, ICML 2004, 2004, pp. 615–622. doi:10.1145/1015330.1015435.

[30] H. and Jerome, Trevor, Tibshirani, Robert, Friedman, The Elements of Statistical Learning The Elements of Statistical LearningData Mining, Inference, and Prediction, Second Edition, 2009. arXiv:arXiv:1011.1669v3, doi:10.1007/978-0-387-84858-7.
URL http://www.worldcat.org/oclc/405547558Hastie,Tibshiranietal-Theelementsofstatisticallearning.pdfhttp://www.springer.com.libproxy1.nus.edu.sg/statistics/statistical+theory+and+methods/book/978-0-387-84857-0http://statweb.stanford.edu/{~}tibs/E

[31] J. H. Holland, Ann Arbor University of Michigan Press 1975, University of Michigan Press, 1975.
URL http://mitpress.mit.edu/catalog/item/default.asp?ttype=2{&}tid=8929

[32] W. Wang, "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, 2017, pp. 422–426. doi:10.18653/v1/P17-2067.

[33] T. Mitra, E. Gilbert, CREDBANK: A large-scale social media corpus with associated credibility annotations, Proceedings of the 9th International Conference on Web and Social Media, ICWSM 2015 (2015) 258–267.

[34] J. Yang, V. Honavar, Feature Subset Selection Using a Genetic Algorithm, in: Feature Extraction, Construction and Selection, Springer US, 1998, pp. 117–136. doi:10.1007/978-1-4615-5725-8_8.

[35] G. Chandrashekar, F. Sahin, A survey on feature selection methods q, Computers and Electrical Engineering 40 (2014) 16–28. doi:10.1016/j.compeleceng.2013.11.024.
URL http://dx.doi.org/10.1016/j.compeleceng.2013.11.024

[36] M. Kudo, J. Sklansky, Comparison of algorithms that select features for pattern classifiers, Pattern Recognition 33 (1) (2000) 25–41. doi:10.1016/S0031-3203(99)00041-2.

[37] F. Horn, R. Pack, M. Rieger, The autofeat Python Library for Automated Feature Engineering and Selection (2019) 1–10arXiv:1901.07329.
URL http://arxiv.org/abs/1901.07329

[38] K. Higgins, Post-truth: A guide for the perplexed, Nature 540 (2016) 9–9. doi:10.1038/540009a.

[39] M. Del Vicario, A. Bessi, F. Zollo, F. Petroni, A. Scala, G. Caldarelli, H. E. Stanley, W. Quattrociocchi, The spreading of misinformation online, Proceedings of the National Academy of Sciences 113 (3) (2016) 554–559. arXiv:https://www.pnas.org/content/113/3/554.full.pdf, doi:10.1073/pnas.1517441113.
URL https://www.pnas.org/content/113/3/554

[40] J. W. Pennebaker, R. L. Boyd, K. Jordan, K. Blackburn, The development and psychometric properties of liwc2015, Tech. rep. (2015).

[41] S. Roy, D. Goldwasser, Weakly supervised learning of nuanced frames for analyzing polarization in news media, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp.

7698–7716.
URL https://www.aclweb.org/anthology/2020.emnlp-main.620

[42] A. Miranda-García, J. Calle-Martín, Yule's characteristic K revisited, Language Resources and Evaluation 39 (4) (2005) 287–294. doi:10.1007/s10579-005-8622-8.

9. Appendix

In the appendix, we list the various resources and features used in this work. For better understanding, we have categorized the features as described in section 4.2, and provided in tables as follows: Table 9.1 shows the dictionaries used, with the features identified, a definition of the feature, and an example; Table 9.2 shows the different complexity and vocabulary richness metrics used, along with their equations; Table 9.3 shows the stylistic features with possible meanings if necessary.

9.1. Dictionary Features

Feature	**Definition**	**Examples**
Loughran Mcdonald Dictionary		
LM_NEGATIVE	Negative tone words	burden, careless
LM_POSITIVE	Positive tone words	advancement, dream, innovator
LM_UNCERTAINTY	Words of uncertainty	approximate, doubted, speculate
LM_LITIGIOUS	Litigious tone words	absolved, crime, executory
LM_CONSTRAINING	Constraining tone words	confines, forbids, unavailability
LM_SUPERFLUOUS	Unnecessary words	assimilate, theses, whilst
LM_INTERESTING	Interesting words	extraordinary, rabbi, toxic
LM_MODAL_STRONG	Strong modal words	always, must, never
Laver Garry Dictionary		
LG_CULTURE_HIGH	Related with high culture	artistic, music, theatre
LG_CULTURE_POPULAR	Related with popular culture	media
LG_CULTURE_SPORT	Related with sport culture	angler, civil war, people
LG_ECONOMY	Related with economy	accounting, earn, loan
LG_ENVIRONMENT	Related with environment	green, planet, recycle
LG_GROUPS_ETHNIC	Related with ethnic groups	Asian, race, ethnic
LG_GROUPS_WOMEN	Related with women	girls, woman, women
LG_INSTITUTIONS_CONSERVATIVE	Related with conservative institutions	authority, inspect, rule

LG_INSTITUTIONS_ NEUTRAL	Related with neutral institutions	chair, scheme, voting
LG_LAW_&_ORDER	Related with law and order	police, punish, victim
LG_RUDAL	Related with countryside	farm, forest, village
LG_VALUES_ CONSERVATIVE	Conservative values	glories, past, proud
LG_VALUES_LIBERAL	Liberal values	cruel, rights, sex
RID Primary Needs		
RID_ORALITY	Orality words	belly, cook, eat
RID_ANALITY	Anality words	anal, dirt, fart
RID_SEX	Related with sex	lover, kiss, naked
RID Primary Sensation		
RID_TOUCH	Related with touching	contact, sting, touch
RID_TASTE	Related with tasting	flavor, savor, spicy
RID_ODOR	Rrelated with smelling	aroma, nose, sniff
RID_GEN_SENSATION	Related with general sensation	awareness, charm, fair
RID_SOUND	Related with sounds	bell, ear, music
RID_VISION	Related with vision	bright, gray, spy
RID_COLD	Related with cold	Alaska, ice, polar
RID_HARD	Related with feels hard in touching	crispy, metal, rock
RID_SOFT	Related with feels soft in touching	feather, lace, velvet
RID Primary Defensive Symbol		
RID_PASSIVITY	Related with passivity	bed, dead, safe
RID_VOYAGE	Related with trips	journey, nomad, travel
RID_RANDOM MOVEMENT	Related with random movements	jerk, spin, wave
RID_DIFFUSION	Related with diffusion	fog, mist, shadow
RID_CHAOS	Related with chaos	char, discord, random
RID_CHAOS	Related with chaos	char, discord, random
RID Primary Regressive Cognition		
RID_UNKNOW	Words for unknown feelings	secret, strange, unknown
RID_TIMELESSNES	Related with infinity time	eternal, forever, immortal
RID_COUNSCIOUS	Words for consciousness alteration	dream, sleep, wake
RID_BRINK-PASSAGE	Words for brink passages	road, wall, door
RID_NARCISSISM	Narcisistic words	eye, heart, hand
RID_CONCRETENESS	Words for something specific	here, tip, wide
RID Primary Icarian Imagery		
RID_ASCEND	Words showing something ascending	climb, fly, wing

RID_DESCENT	Words showing something descending	dig, drop, fall
RID_HEIGHT	Related with height	bird, hill, sky
RID_DEPTH	Related with depth	cave, hole, tunnel
RID_FIRE	Related with fire	solar, coal, warm
RID_WATER	Related with water	ocean, sea, pool
RID Secondary Feeling		
RID_ABSTRACT_ TOUGHT	Related with abstraction	know, may, thought
RID_SOCIAL_ BEHAVIOR	Related with social behavior	ask, tell, call
RID_INSTRU_ BEHAVIOR	Related with instrumental behavior	make, find, work
RID_RESTRAINT	Related with restraint behavior	must, stop, bind
RID_ORDER	Related with order(form)	measure, array, system
RID_TEMPORAL_ REPERE	Related with temporal references	when, now, then
RID_MORAL_ IMPERATIVE	Related with moral imperatives	should, right, virtue
RID Emotions		
RID_POSITIVE_ AFFECT	Related with positive emotions	cheerful, enjoy, fun
RID_ANXIETY	Related with anxiety emotions	avoid, horror, shy
RID_SADNESS	Related with sad emotions	hopeless, pain, tragic
RID_AFFECTION	Related with affection	bride, like, mercy
RID_EXPRESSIVE_ BEH	Related with expressive behavior	dance, sing, art
RID_GLORY	Related with glory	elite, kingdom, royal
RID_GLORY	Related with glory	elite, kingdom, royal
AFINN Dictionary		
AFINN score	The AFINN lexicon is a list of English terms manually rated for valence with an integer between -5 (negative) and +5 (positive) by Finn Årup Nielsen	abuses: -3, amazing: 4, avoid: -1

Table 7: Dictionary Features

9.2. Complexity Features

Feature	Definition
Readability Index	
Flesch reading ease	$$206.835 - 1.015\left(\frac{total\#ofwords}{total\#ofsentences}\right) \quad (3)$$

Flesch–Kincaid	$$0.39\left(\frac{total\#ofwords}{total\#ofsentences}\right)+11.8\left(\frac{total\#ofsyllables}{total\#ofwords}\right)-15.59 \quad (4)$$
SMOG	$$1.0430\sqrt{\#ofpolysyllables*\frac{30}{\#ofsentences}}-15.59 \quad (5)$$
Automated readability index	$$0.39\left(\frac{total\#ofwords}{total\#ofsentences}\right)+11.8\left(\frac{total\#ofsyllables}{total\#ofwords}\right)-15.59 \quad (6)$$
Dale-Chall	$$0.1579\left(\frac{difficultwords}{total\#ofwords}*100\right)+0.0496\left(\frac{total\#ofwords}{total\#ofsentences}\right) \quad (7)$$ *Dale-Challe declare a list with difficult words
Coleman–Liau	$$0.0588L-0.296S-15.8 \quad (8)$$ *L = Total # of Letters / Total # of Words * 100* *S = Total # of Sentences / Total # of Words * 100*
Gunning fog	$$0.4\left[\left(\frac{Total\#ofwords}{Total\#ofsentences}\right)+100\left(\frac{Total\#ofcomplexwords}{Total\#ofwords}\right)\right] \quad (9)$$
Vocabulary Richness	
Yule K	Miranda-Garcia et al. [42]
TTR	$(Total\#ofuniquewords/Total\#ofwords)*100$
Brunets Index	$N^{V^{-a}}$, where N is the text length, V is the number of unique words, and –a is a scaling constant that is usually set at –0.172
Sichel	$Total\#ofhappaxdislegomena/Total\#ofwords$

Table 8: Complexity Features

9.3. Stylistic Features

Feature	Meaning
Part Of Speech Tags	
CC	Coordinating conjunction
CD	Cardinal digit
DT	Determiner
EX	Existential there (like: “there is” . . . think of it like “there exists”)
FW	Foreign word
IN	preposition/subordinating conjunction
JJ	adjective ‘big’

JJR	adjective, comparative ‘bigger’
JJS	adjective, superlative ‘biggest’
LS	list marker 1)
MD	modal could, will
NN	noun, singular ‘desk’
NNS	noun plural ‘desks’
NNP	proper noun, singular ‘Harrison’
NNPS	proper noun, plural ‘Americans’
PDT	predeterminer ‘all the kids
POS	possessive ending parent’s
PRP	personal pronoun I, he, she
PRP$	possessive pronoun my, his, hers
MD	modal could, will
RB	adverb very, silently
RBR	adverb, comparative better
RBS	adverb, superlative best
RP	particle give up
TO,	to go ‘to’ the store.
UH	interjection, errrrrrrrm
VB	verb, base form take
VBD	verb, past tense took
VBG	verb, gerund/present participle taking
VBN	verb, past participle taken
VBP	verb, sing. present, non-3d take
VBZ	verb, 3rd person sing. present takes
WDT	wh-determiner which
WP	wh-pronoun who, what
WP$	possessive wh-pronoun whose
WRB	wh-abverb where, when

Table 9: Part Of Speech Features

Feature	**Meaning**
Structural	
total_number_of_sentences	
total_number_of_words	
total_number_of_characters	
total_number_of_begin_upper	Words with first capital letter
total_number_of_begin_lower	Words with first lowercase letter
total_number_of_all_caps	Word with all capital letters
total_number_of_stopwords	
total_number_of_lines	

number_of_I_pronouns	
number_of_we_pronouns	
number_of_you_pronouns	
number_of_he_she_pronouns	
number_of_exclamation_marks	
number_of_quotes	
number_of_happax_legomena	Word types that occur only once in text
number_of_happax_dislegomena	Word types that occur only twice in text
has_quoted_content	
ratio_alphabetic	
ratio_uppercase	
ratio_digit	
avg_number_of_characters_per_word	
avg_number_of_words_per_sentence	
avg_number_of_characters_per_sentence	
avg_number_of_begin_upper_per_sentence	
avg_number_of_all_caps_per_sentence	
avg_number_of_begin_lower_per_sentence	
avg_number_of_stopwords_per_sentence	

Table 10: Structural Features