Web-based Startup Success Prediction

Boris Sharchilev*
Yandex
Moscow, Russia
bshar@yandex-team.ru

Michael Roizner
Yandex
Moscow, Russia
roizner@yandex-team.ru

Andrey Rumyantsev
Yandex
Moscow, Russia
arumyan@yandex-team.ru

Denis Ozornin
Yandex
Moscow, Russia
dozornin@yandex-team.ru

Pavel Serdyukov
Yandex
Moscow, Russia
pavser@yandex-team.ru

Maarten de Rijke
University of Amsterdam
Amsterdam, The Netherlands
derijke@uva.nl

ABSTRACT

We consider the problem of predicting the success of startup companies at their early development stages. We formulate the task as predicting whether a company that has already secured initial (seed or angel) funding will attract a further round of investment in a given period of time. Previous work on this task has mostly been restricted to mining structured data sources, such as databases of the startup ecosystem consisting of investors, incubators and startups. Instead, we investigate the potential of using web-based open sources for the startup success prediction task and model the task using a very rich set of signals from such sources. In particular, we enrich structured data about the startup ecosystem with information from a business- and employment-oriented social networking service and from the web in general. Using these signals, we train a robust machine learning pipeline encompassing multiple base models using gradient boosting. We show that utilizing companies' mentions on the Web yields a substantial performance boost in comparison to only using structured data about the startup ecosystem. We also provide a thorough analysis of the obtained model that allows one to obtain insights into both the types of useful signals discoverable on the Web and market mechanisms underlying the funding process.

CCS CONCEPTS

• **Information systems** → **Web mining**; *Decision support systems*;

KEYWORDS

Predictive modeling, Heterogeneous web data, Mining open sources, Gradient boosting

ACM Reference Format:
Boris Sharchilev, Michael Roizner, Andrey Rumyantsev, Denis Ozornin, Pavel Serdyukov, and Maarten de Rijke. 2018. Web-based Startup Success Prediction. In *Proceedings of CIKM 2018*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.475/123_4

*Also with the University of Amsterdam.

CIKM 2018, October 22–26, 2018, Lingotto, Turin, Italy

ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/10.475/123_4

1 INTRODUCTION

Startup companies form a crucial part of the modern world's economic infrastructure. For example, young and small companies have higher job creation rates than large and mature firms [2]. New ventures drive technological innovation and development by injecting competition into their field and bringing innovative ideas to the table. These factors make startup companies an important target for analysis. Startups operate in an extremely risky and competitive environment. On average, only around 60% of new companies stay in business for more than three years [14]. An important factor determining a venture's ability to survive is managing to gather enough funding for business sustainability and extension. Identifying successful startups during their early development stages is highly beneficial both for ventures to identify improvement opportunities and for the investors backing them to be ahead of the competition—this is the task we set ourselves in this paper.

Despite obvious practical importance, the problem lacks exposure both from the research community and the financial industry. From the industrial perspective, several venture funds like Google Ventures[1] or PreSeries[2] heavily rely on algorithms to assist their decisions with huge success. However, the mechanics of their algorithms are private and heavily protected, and openly available research efforts to reproduce and advance their progress would be of great value to the research community. In the academic literature, there have been several previous attempts to approach the problem of identifying successful startups, e.g., in the setting of merger and acquisition (M&A) prediction [29] or portfolio optimization [32]. These approaches generally proceed by applying a chosen machine learning technique to structured data on startups extracted from a specialized database about the startup ecosystem consisting of investors, incubators and startups, such as Crunchbase[3] and VentureSource.[4] These approaches are limited both in terms of the data they use and the learning methods they employ.

We seek to address both opportunities for improvement. On the conceptual side, we propose to expand structured data about the startup ecosystem with a rich source of signals not covered by previous research: a large volume of diverse types of company mentions gathered from the open web. The intuition behind this strategy is that a company or product that stands out from its competition will resonate with its target audience, and information about this will be discoverable on the open web. On the technical side, we suggest a robust and diversified machine learning workflow, Web-Based

[1] https://www.gv.com/
[2] https://preseries.com/
[3] https://www.crunchbase.com/#/home/index
[4] https://www.dowjones.com/products/pevc/

Startup Success Prediction (WBSSP), capable of leveraging our large, multi-modal set of signals available for learning. Importantly, WBSSP is implemented using an open source gradient boosting library and all data used in this study is collected from public sources, which only requires other researchers to have access to a simple web crawler to reproduce our results.

Specifically, we address the problem of predicting funding events for startups, that is, whether a startup that already received an initial round of funding will be able to secure a next level of investment within a predefined time horizon. Our signals include basic company data, history and track record of its investors, founders and staff, textual descriptions, news articles and other types of mentions discovered on the web, etc. These signals are mapped to two types of features: *dense* features that can be directly fed to a classifier without inflating the feature space and, unlike in previous related work, *sparse* features whose dimensionality is first reduced prior to being used in our predictive pipeline. Our learning approach consists of several overfitting-robust models distilling information from sparse features, which, along with other "naturally" dense features, are fused using CatBoost, a state-of-the-art modification of Gradient Boosted Decision Trees (GBDT) [6].

Our contributions in this paper are four-fold:

(1) We demonstrate how to successfully use web-based open source for the problem of startup success prediction.
(2) We conduct the largest experiments to date on the problem of startup success prediction in terms of magnitude and diversity of the utilized data.
(3) We set up a pipeline for learning from heterogeneous company-related data based on Gradient Boosting, Logistic Regression (LR) and a Neural Network (NN) that is able to achieve better than state-of-the-art performance.
(4) Our results provide evidence for the "wisdom of the crowd" paradigm previously noted in social studies [27] and, in the context of the "the Jockey or the Horse" dilemma [16] concerning the relative importance of management and business value, for betting on the "horse," that is, on the business.

2 MOTIVATION AND RELATED WORK

2.1 Motivation

In order to properly formulate the startup success prediction problem, the notion of "success" has to be formalized in a meaningful way. The definition should satisfy two main conditions: First, it should translate to real profitability. Second, success defined that way should be both available for evaluation (that is, it should be determinable from publicly available data) and should not require us to forecast into the distant future, in order to maintain tractability.

2.1.1 Revenue. A perfect success metric would be *revenue*. Generating revenue is the ultimate financial goal of a company, and this is what investors actually expect when allocating funding. Unfortunately, this is a difficult target for prediction: first, revenues do not have to be disclosed and, thus, are not public information in general. Second, it may take up to eight years for an average company to become profitable [3]. Thus, we turn to selecting a suitable investor-company interaction to predict.

2.1.2 M&A. One such type of interaction is an *M&A (Merger and Acquisition)* event. The fact of a particular company being acquired usually demonstrates the acquiring party's high regard of the company's business. A downside to this approach is that M&A prediction is an imperfect proxy metric for success both in terms of precision and recall: not all successful companies get acquired and, importantly, only a fraction of acquired companies become successful and yield financial returns to their shareholders [23]. Moreover, M&A motivations can be unfavorable from a revenue-seeking investor's viewpoint; think, e.g., of *acqui-hire deals* [8].

2.1.3 Funding events. Instead of predicting M&A processes, the choice we make in this paper is to focus on predicting *funding rounds* attracted by a startup. Much like with M&A, the fact of a startup securing funding is a strong indicator of its current or potential business value, as evaluated by an investor, a highly informed expert in the field [9]. A convenient trait of predicting types of attracted funding rounds is the flexibility of this approach: by changing the type of "target" round we can balance the amount of risk versus potential reward sought by an investor. See Section 3.

2.2 Related work

2.2.1 Finance and economics. Understanding the mechanics of angel, venture capital and private equity investment processes and motivations of both investors and ventures is a problem of great importance in economics and finance and, thus, has attracted significant attention by researchers in these fields; see, e.g., [9, 18, 28]. These publications focus on analyzing various financial aspects of the problem and do not aim at building an automated predictive model. The most relevant body of work of this type investigates either objective reasons for companies' successes and failures or reasoning behind investors' decisions to provide or deny funding, e.g., [1, 10, 12, 13, 16, 17, 21, 22]. These studies provide valuable insights into what types of data should be used and what kind of signals should be extracted from it.

Despite our deliberate limitation to open web sources, analyses based on our predictive model provide empirical evidence in support of (or contradicting) some of the results from the studies listed above. In particular, our work falls in line (1) with [1, 13], where blogger opinions and/or news are found to be correlated with a company's success at some stage of the funding process; (2) with [27], where the "wisdom of the crowd" paradigm is being studied, noting the superiority of aggregated judgments of a group over an individual expert; and (3) with [16], where, perhaps surprisingly, a company's market, investors and business idea quality are found to be more important for eventual success than the expertise of the original founding team (the so-called "Jockey or the Horse" dilemma).

2.2.2 Data mining and machine learning. In contrast to the financial literature, the problem of startup success prediction has been little studied in terms of predictive modeling and machine learning. Several papers approach the problem from a very narrow angle of a particular industry [20] or country [11]. Compared to our work, these publications are severely constrained, both in terms of the scale of the data used and in terms of the predictive tools used. Another relevant study, [31], only considers user engagement data from social media, in contrast to the analysis of the full range of web mentions that we utilize.

Several publications consider alternative, orthogonal choices of modeling the startup success prediction problem, such as portfolio optimization [26, 32] or link prediction [19, 30]. Publications of this type attempt to solve a much more uncertain problem than direct discriminative success prediction considered in our work, because they try to either predict a startup-investor pair instead of just a successful startup (link prediction) or to also take other funded companies into account (portfolio optimization); this fundamental uncertainty takes a toll on predictive quality.

The most relevant related study is [29], which deals with predicting a proxy for company success, in their case, M&A deals, by training a classifier on data gathered from Crunchbase. However, this work has several serious limitations. First, it appears to be prone to using "leaked" information from after the prediction date, e.g., *#employees*, the historical values of which are not tracked, and the number of profile revisions, for which only the date of the last edit is available per each contributor. Second, it is restricted almost exclusively to Crunchbase data, while our study enriches it with a large body of diverse and openly available data from both LinkedIn and the web in general. Third, apart from topic model features, Xiang et al. [29] only use aggregated dataset statistics for prediction, whereas we also learn from much richer (*sparse*) data representations, like individual company investors and domains mentioning a particular startup. Finally, in contrast to a simple Bayesian Network classifier utilized by Xiang et al. [29], we develop a robust and diversified machine learning pipeline, WBSSP, including Logistic Regression, a Neural Network and a state-of-the-art GBDT modification, CatBoost [6].

3 PROBLEM STATEMENT

We focus on predicting *funding events*. An appealing trait of this formulation of startup success is its flexibility in balancing investment risk and promptness in discovering startups: funding events are usually classified into *rounds*[5] of increasing magnitude both of investments and participating companies, ranging from initial angel [28] and seed rounds to series A/B/C and onwards, involving giants like Google or Facebook [9]. The larger the funding round, the more established a company is and the more information is available to base the prediction on.

This sets up a convenient framework for balancing risks and possible rewards that we wish to undertake in our prediction: we only consider companies that have already reached a certain type of funding round (the *trigger round*) as candidates, and predict whether they will secure a funding round of another type (the *target round*) in a given amount of time (*horizon*). We choose angel and seed rounds as triggers, all further rounds (Series A onwards) as targets, and fix the horizon to be one year. As shown in Table 3 (*Companies*), post-seed funding is a selective process, with only about 11% of seed-funded companies eventually securing a Series A+ round, which highlights the business relevance of our formulation.

In summary, our predictive problem is formulated as follows: *for a given startup that has received seed or angel funding, predict whether it will secure a further Series A or larger round of funding during the next year.*

4 APPROACH

First, we define the subjects and targets of our predictions; after that, we specify the sources of data used for prediction and the features that we extract from it, along with motivations why they may be useful; finally, we conclude this section by detailing the prediction pipeline and machine learning algorithms used for training.

[5] https://en.wikipedia.org/wiki/Venture_round#Round_names

4.1 Sample and targets

We have already motivated predicting the fact of attracting a next round of funding in a given time horizon, after having secured initial funding. *When* exactly to make a prediction is still an open question. Here, two different general approaches are possible:

- *Company-centric*: for each company, make predictions n days after seed funding.
- *Investor-centric*: fix a date t and make a prediction at this date for each candidate startup.

We adopt an investor-centric approach. In addition to more closely resembling a real-world investment use case, it also has a *data augmentation* side effect: each company may be reused multiple times during training and testing, effectively increasing the dataset by an order of magnitude in comparison to a company-centric approach; see Table 3 for the exact numbers. We also note several notable aspects of our setup: first, startups may be at different development stages and, thus, it may be easier to make an accurate prediction for an older company versus a younger one that just got seed funding. Second, after we split our data into training and test sets by a certain date (see Section 5.2), a particular company's snapshots taken at different moments may be present both in the training and test sets; this is not a leakage since a startup's features and prediction target change over time. It is important to note that these aspects of our setup are intentional, since the training and testing scenarios exactly mirror **the actual intended use-case** of the model in **a real investment decision-making process**.

Our predictive model has to be capable of making forecasts at different time moments for each company. Thus, we construct our training and test sets by sampling multiple prediction dates, extracting corresponding "snapshots" of startups that were candidates at the moment (as defined in Section 3). Furthermore, at each prediction date, we only consider startups that had a trigger round during the past year in order to filter out "stale" companies. The algorithm for training/test set construction is given in Algorithm 1. More details about the construction process and particular values of Algorithm 1's parameters are given in Section 5.2.

4.2 Features

In this section, we describe the features that we use for training our model. They can be classified into four broad categories according to the information sources that they capture: *general*, *investor*, *people*, and *mentions*; see Table 1 for an overview and the exact listing.

4.2.1 General features. This group encompasses the most basic information about a company that gives a general idea of where the company currently stands. They include: a startup's country(-ies) of operation, HQ location, industry, age, textual description etc.

4.2.2 Investor features. Factors from this category capture the information about the startup's history of funding and engagement with investors. Backing by a strong investor is, intuitively, correlated and even causally related to a venture's success [10], so we expect these features to significantly influence the prediction quality. Investor features include: number and types of previously secured funding rounds, amounts of investments attracted on each round, statistics of previous investors that reflect their historical success both in a specific industry and globally, etc.

4.2.3 People features. While a company's funding history reflects the external evaluation of a venture's potential, it is the company's

Table 1: Features used. Legend: *Final?* indicates whether a feature is directly given to the final GBDT classifier; *LR group* is the Logistic Regression group number used in Section 4.3 (“×" means “not used by LR"); ∞ means an unconstrained/dataset-dependent number of features; $\{option\}$ means enumeration of all $option$ values; $(option\_a/option\_b)$ denotes variations of similar features; $(?option)$ denotes an optional feature name modifier.

Group	Subgroup	Name	Description	Type	Number	Sparse?	Final?	LR group
General	Age	*age*	Days since foundation	Numeric	1	×	✓	1
		year_thresholds	Indicators $1_{year>t}$, t = 2000…2017	Flags	21	✓	×	1
	Industry	*categories*	Company's Crunchbase (CB) categories	Flags	732	✓	×	2
		categories_count	Number of CB categories	Numeric	1	×	✓	2
		competitors	Company's competitors on CB	Flags	∞	✓	×	3
		competitors_count	Number of competitors on CB	Numeric	1	×	✓	3
	Websites	*websites*	Are Facebook/Twitter/LinkedIn/homepage on CB?	Flags	4	✓	×	4
		websites_count	Number of websites listed on CB	Numeric	1	×	✓	4
		websites_ _created_m(6/12/24)	Number of websites created in last 6/12/24 months	Numeric	3	×	✓	×
	Offices	*(offices/hq)*	Numbers of (?HQ) offices in different countries or, if available, cities	Numeric	∞	✓	×	5
		(offices/hq)_count	Number of (?HQ) offices	Numeric	2	×	✓	5
		(offices/hq)_(min/max/avg)_age	Statistics (min/max/average) of ages of (?HQ) offices	Numeric	6	×	×	5
	Description	*(?short_)description*	Textual description, bag-of-words	Numeric	∞	✓	×	6
	Products	*products_count*	Number of products	Numeric	1	×	✓	7
		products_(min/max/avg)_age	Statistics of ages of products	Numeric	3	×	×	7
Investor	Investor-level	*investors*	Number of investments made by each investor	Numeric	∞	✓	×	8
		investor_money	Money invested by each investor	Numeric	∞	✓	×	9
		investor_shares	Same but normalized by total raised money	Numeric	∞	✓	×	10
	Round-level	*funding_types*	Counts of funding types as given in CB, e.g. *seed*, *angel*, *venture* etc.	Numeric	8	✓	×	11
		funding_types_money	Money raised in different funding types	Numeric	8	✓	×	11
		currencies	Numbers of rounds funded in different currencies	Numeric	39	✓	×	11
	Aggregates	*round_count*	Number of secured funding rounds	Numeric	1	×	✓	11
		investment_count	Number of investments received so far	Numeric	1	×	✓	11
		total_money	Total money raised so far	Numeric	1	×	✓	11
		money_unknown	Number of rounds without valuations	Numeric	1	×	✓	11
		round_(min/max/avg)_age	Statistics of times since past rounds	Numeric	3	×	×	11
		investor_count	Number of past investors	Numeric	1	×	✓	8
		investor_(min/max/avg)_time	Statistics of times since investors got involved with the company	Numeric	3	×	×	8
		seed_money_raised	Money raised on seed round(s)	Numeric	1	×	✓	×
		investor_ _{round_a}_{round_b}_ _(?cat_) (sum/max)_(sum/max)	For each startup's investor i and each company c, calculate $s_{ic} = 1_{c \text{ got } round\_a} \cdot 1_{c \text{ got } round\_b} \cdot$ $\cdot 1_{i \text{ invested in } c}$. Aggregate over rows and columns with sum/max, sum/max. (20 most important combinations are used in final model.)	Numeric	512	✓	✓	12
	Own investments	*own_investments*	Numbers of investments made in each company	Numeric	∞	✓	×	13
		own_investments_count	Total number of investments made	Numeric	1	×	✓	13
Team	Board	*board*	IDs of board members	Flags	∞	✓	×	14
		board_(?(male/female)_)count	Number of all/male/female board members	Numeric	3	×	✓	14
		board_(min/max/avg)_time	Statistics of members' times on the board	Numeric	3	×	×	14
	Founders	*founders*	IDs of company founders	Flags	∞	✓	×	15
		founders_(?(male/female)_)count	Number of all/male/female founders	Numeric	3	×	✓	15
		founders_ _{round_a}_{round_b}_ _(?cat_) (sum/max)_(sum/max)	Analogous to similar investor features	Numeric	512	✓	✓	16
	Team (CB)	*team*	IDs of current team members on CB	Flags	∞	✓	×	17
		team_(?(male/female_)count	Number of all/male/female members	Numeric	3	×	✓	17
		team_(min/max/avg)_time	Statistics of members' times on the team	Numeric	3	×	×	17
		team_(current/created/ started/ended)_m(6/12/24)	Number of current/CB registered hired or released staff in the last 6/12/24 months	Numeric	12	×	✓	×
	LinkedIn	*linkedin_jobs*	Number of people with a given job title and company in LinkedIn resume. (All job titles used as features.)	Numeric	∞	✓	×	18
Mentions	News (CB)	*news_count*	Number of CB news articles	Numeric	1	×	✓	19
		news_(created/posted)_ _m(6/12/24)	Number of CB news items added to CB posted in last 6/12/24 months	Numeric	6	×	✓	×
		news_domains	Counts of mentions on each domain	Numeric	∞	✓	×	19
		news_tm	Topic model (LDA) features	Numeric	5	✓	×	20
	Links	*links_(domains/references)_ _(?log_)(total/m6/m12/m18)*	(Logarithm of) number of domains/pages mentioning the company in total/last 6/12/18 months	Numeric	16	×	✓	×
		links_domains_ _(flag/linear/log)_(total/m6)	IDs/counts/log of counts of mentions on each domain in total/in last 6 months	Numeric/Flags	∞	✓	×	21

Algorithm 1 Training/test set construction

```
function PopulateSample(companies, train_start, train_end, test_start, test_end, step, trigger_rounds, target_rounds)
    train_sample, test_sample ← [], []
    date ← train_start
    while date < test_end do
        candidates_at_date ← CandidatesAtDate(companies, date, trigger_rounds, target_rounds)
        if date < train_end then
            train_sample.extend(candidates_at_date)
        else if date ≥ test_start then
            test_sample.extend(candidates_at_date)
        end if
        date ← date + step
    end while
     return train_sample, test_sample
end function
function CandidatesAtDate(companies, date, trigger_rounds, target_rounds)
    candidates ← []
    targets ← []
    for company in companies do
        if HadRoundTypeLastYear(company, date, trigger_rounds) then
            features ← GetFeaturesAtDate(company, date)
            candidates.append((company_id, date, features))
            targets.append(HadRoundTypeLastYear(company, date + 365, target_rounds))
        end if
    end for
     return candidates, targets
end function
```

team that drives its development internally. In addition to Crunchbase data, to incorporate fine-grained information about staff experience, we crawled a large number of LinkedIn[6] profiles and incorporated this information for people who specified their LinkedIn profiles on Crunchbase. Specific features of this group include: number of founders, statistics of their past ventures' successes (if any), experience of a startup's employees in the past etc.

4.2.4 Mentions. Importantly, in addition to Crunchbase and LinkedIn, we also consider a data source that has not been studied so far: a detailed crawl of a startup's presence on the web. A fraction of mentions is also indexed by Crunchbase in the form of news articles; however, we note that our crawl is considerably broader than Crunchbase's data in several ways, both in scale (see Fig. 1) and quality: Crunchbase mostly indexes news and/or analytics from well-established tech and finance media outlets. However, our dataset is not limited to such "clean" mentions; some examples are given in Table 2. They range from articles on major financial news websites to commentaries on specialized discussion boards; see Section 6.3 for examples of significant domains. The use of this type of data is motivated by the "wisdom of the crowd" [27] paradigm stating that aggregation of a large set of opinions and/or ideas from a large group of individuals tends to lead to better insights or predictions than given by individual experts. Specifically, we calculate both aggregated statistics of a startup's online presence (total number of mentions in the last 6/12/18 months, unique domains mentioning a startup, etc.) and individual mentions of a company on different domains. As shown in Section 6, these features are already among the strongest predictors overall, which is a research finding on its own.

[6] https://www.linkedin.com

4.3 Learning algorithm

We now turn our attention to the learning pipeline of our approach, WBSSP; see Fig. 2 for a schematic overview. The final element of WBSSP is CatBoost, a state-of-the-art GBDT modification [6]. This choice is motivated by CatBoost's robustness in treating different types of data and its superior classification performance.[7] It is trained on features from groups defined in Table 1. Our features consist of two classes that have to be treated differently: *dense* features like aggregated investor or mention statistics represent data that can be directly fed to a classifier without inflating the feature space and introducing overfitting issues. On the other hand, because of their large dimensionality, using *sparse* features directly without proper regularization will lead to severe overfitting. Thus, to pass sparse feature information in a condensed way to the downstream classifier, we first train two robust models capable of dealing with such data, Logistic Regression and a Neural Network (NN), and use predictions of these models as extra features for the final classifier.

First, we train an L_2-regularized LR on the combination of sparse and dense features. We train LR in an "online" fashion [5] by re-training the model each N days and, for each sample, using the "freshest" model trained so far. LR features are semantically grouped as described in Table 1, "LR group" column; for each startup, we then extract both total and individual LR feature group scores; that is, if $F_i = \{f_{i_k}\}_{k=1}^{N_i}$, $i = 1, \dots, N_g$ are the LR feature groups, and $p(y = 1 \mid x) = \sigma(w^T x)$ is the trained LR model, we calculate the i-th group's score for object x as $Score_i(x) = \sum_{k=1}^{N_i} w_{i_k} x_{i_k}$.

Second, we aim to further exploit the signal captured by our crawled startup mentions. It is desirable to utilize a model that is both capable of learning non-linear relationships and robust to overfitting. To that end, we train a neural network (NN) on features from the Mentions group (Section 4.2.4). The NN architecture we use has two fully connected hidden layers of 128 neurons with ReLU [24] nonlinearities, where each is followed by a batch normalization layer [15] and a dropout [25] layer with rate 0.8 for heavy regularization. The training set is split into 10 folds, and out-of-fold predictions are used for the downstream classifier. However, preliminary experiments showed that training directly on the numbers of mentions per domain leads to noisy results; the intuition is that, for each domain, what essentially matters is the qualitative

[7] See http://www.catboost.yandex, *Benchmarks* section.

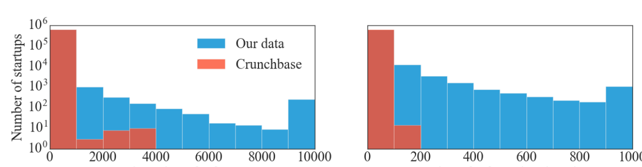
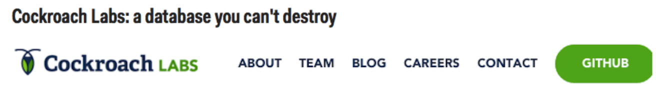
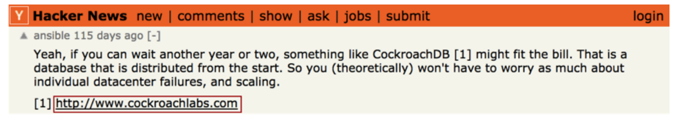


Figure 1: Histogram of the number of mentions and unique mentioning domains per company that are captured by Crunchbase and our web crawl.

Table 2: Examples of mentions of a particular startup, CockroachDB. Top row: article on a major news portal; middle row: professional discussion on a dedicated forum; bottom row: entry on a wiki page of a popular software project.

Domain	Title	Mention
businessinsider.in	CockroachDB: A database you can't destroy	
news.ycombinator.com	RethinkDB versus PostgreSQL: my personal experience	
github.com	Sites using React	

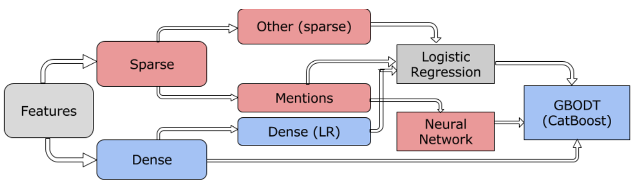


Figure 2: Schematic depiction of our prediction pipeline, WBSSP. Blue indicates dense features or blocks, red indicates sparse, and gray indicates a mixture of both.

characteristic of the number of mentions (e.g., *none*, *a few* or *a lot*) and not necessarily the exact count c. Thus, we predefine a set of exponentially increasing threshold values $t_1, \ldots, t_m$ and, for each startup-domain pair, calculate a set of sigmoids $\{\sigma(c - t_i)\}_{i=1}^{m}$, which are smoothened versions of indicator functions $\{1_{c>t_i}\}_{i=1}^{m}$.

5 EXPERIMENTAL SETUP

5.1 Research questions

We seek to answer the following research questions: **(RQ1)** How does WBSSP compare to the current state-of-the-art? **(RQ2)** Does WBSSP's joint treatment of dense and sparse features help improve upon the approach which only uses dense aggregate features? **(RQ3)** What types of signal make the largest contributions to the model? To what extent do startup mentions on the open web contribute to prediction quality? **(RQ4)** Is the magnitude of a startup's web presence sufficient for success prediction, or does learning the importance of particular sources of mentions matter as well? **(RQ5)** Do company mentions become stronger success predictors when aggregated at a larger scale? That is, does the "wisdom of the crowd" work for our predictive problem?

5.2 Dataset description

For our main source of data, we crawl Crunchbase, one of the largest databases on public and privately held companies, up until May 2017. Specifically, we download all of the data from the *organizations*, *funding_rounds* and *people* endpoints.[8] We train on startup snapshots up until May 2014 and test on snapshots dated May 2015 to May 2016, captured with a step of 30 days between the closest snapshots. We also enrich our data with a crawl of people profiles from LinkedIn, dated March 2017.

For monitoring a company's web presence, we utilize a detailed crawl of the observable web used in building the web index of Yandex,[9] a major Russian search engine. This data comes in the form of a *web graph*, where each node is a URL of a web page and each (directed) edge is a hyperlink. If a web page is connected to a company's website specified on Crunchbase, we consider it to be *mentioning* that company.[10] Moreover, we only use web pages with unambiguous publication dates extracted by a proprietary dating algorithm. We specifically note that all of the crawled pages are openly discoverable and indexed by search engines, which makes our results reproducible by using a simple web crawler.

We construct the training/test samples and feature representations as described in Algorithm 1 and Table 1, respectively. Statistics for both companies and learning samples are given in Table 3.

Table 3: Main dataset statistics. "Positive class" for a company means that it eventually secured a target funding round.

	Training set		Test set	
	Total	Positive class	Total	Positive class
Companies	21,947	2,912	15,128	1,206
Samples	224,708	22,478	91,477	6,441

5.3 Baselines

We now describe the baselines used for comparison. To study how different feature groups influence the model, we use the following:

[8] https://data.crunchbase.com/
[9] https://www.yandex.ru/
[10] For *links_domains_** features from Table 1, to maintain tractability, we only use the top-10000 domains having the most mentions.

Random Samples binary success predictions from prior success distribution estimated from the train labels.

General (+Inv[estor] (+ Team (+ Sparse))) Discards LR and NN and only uses dense features from the corresponding groups (see Table 1) for training CatBoost. *Sparse* also adds LR and NN features from the General/Investor/Team groups.

No Domains Adds dense aggregates from *Mentions* group to *General + Inv + Team + Sparse*. The only difference with WBSSP is that sparse *Mentions* features are not included.

Next, to compare WBSSP to the state of the art, we implement

SOTA (State-of-the-art) This baseline is based on the approach of [29], which was originally used to predict M&A events. Although exact feature design and machine learning algorithm used are not the same, we still capture all of the "non-leaking" groups of signals that were considered in that study. Like previous baselines, *SOTA* also uses a single classifier trained only on a set of dense features. However, to simplify comparison with other baselines, we strengthen *SOTA* by using the state-of-the-art GBDT algorithm instead of a simple Bayesian Network classifier. For *Mentions*, following [29], *SOTA* only includes news from TechCrunch.[11] Moreover, as in [29], we also train LDA [4] with 5 topics on TechCrunch news headlines and use a company's topic profile as extra features.

5.4 Evaluation metrics

We now describe the metrics that we use to evaluate the quality of our predictions. First, we use **ROC-AUC**, a standard classification metric. Second, for a clear measure of performance quality from a business perspective, we analyze the Precision-Recall (PR) curve. In a practical scenario an investor will only be able to fund a very small fraction of startups, so our interest lies with the low-recall region of the curve. To formalize this intuition, we also consider lists of top-100 and top-200 companies (ordered by success probability predicted by our method) and, for these lists, calculate Precision and F_β scores ($\beta = 0.1$ to stress greater importance of precision over recall for our evaluation). We denote them as **P@k** and $\mathbf{F_{0.1}@k}$, $k = 100, 200$, respectively.

For significance testing, we bootstrap the test set, measure performance metrics for each bootstrapped sample and use a one-sided Wilcoxon signed-rank test ($^{***}\ p < 0.01$; $^{**}\ p < 0.05$; $^{*}\ p < 0.1$).

6 RESULTS

6.1 Success prediction quality

6.1.1 Metrics. We train WBSSP and the baselines described in Section 5.3 on the training set and report the obtained quality metrics on the test set. Results are given in Table 4 and Fig. 3.

Table 4: Performance metrics for classifiers trained on different feature groups.

Features	P@100	$F_{0.1}$@100	P@200	$F_{0.1}$@200	ROC-AUC
Random	0.059	0.049	0.030	0.046	0.500
General	0.100	0.068	0.095	0.080	0.615
General + Inv	0.250	0.166	0.305	0.260	0.800
General + Inv + Team	0.310	0.203	0.325	0.286	0.798
General + Inv + Team + Sparse	0.455	0.278	0.465	0.355	0.803
No Domains	0.410	0.258	0.420	0.329	0.807
SOTA	0.270	0.209	0.335	0.298	0.800
WBSSP	**0.626*****	**0.383*****	**0.535*****	**0.439*****	**0.854*****

[11] https://techcrunch.com/

As can be seen from Table 4, WBSSP outperforms all of the compared methods; in particular, it increases ROC-AUC by 6.75%, P@100 by 131.9% and $F_{0.1}$@100 by 83.3% over *SOTA*. The differences between WBSSP and all other approaches (including *SOTA*) are statistically significant.

Also, Fig. 3 shows a huge advantage of WBSSP over the baselines. For example, at recall level of 5%, the success rate is higher than 60%, in contrast to about 40% for *SOTA*. These two facts unambiguously answer **RQ1** in favor of WBSSP. To avoid possible confusion, we also note that 60% precision is not only a significant relative advantage over the current *SOTA*, but also an objectively strong result for the problem at hand: for example, an "uninformative" random baseline yields approximately 6% precision, upon which WBSSP improves ten-fold.

6.1.2 Sparse features contribution. We also separately analyze the benefits of learning from sparse signals, both from structured Crunchbase data (*General + Inv + Team* vs. *General + Inv + Team + Sparse*) and company mentions discovered on the web (*No Domains* vs. *WBSSP*). From Table 4 we see that WBSSP's treatment of sparse features is helpful for both data sources, improving P@100 by 46.8%, $F_{0.1}$@100 by 36.9% in the former case and ROC-AUC by 5.8%, P@100 by 52.7%, $F_{0.1}$@100 by 48.4% etc. in the latter.

First, these results show that WBSSP's fusion of multiple models does indeed give a huge performance boost over simply learning from aggregated dense features, allowing us to answer **RQ2** positively. Second, WBSSP's superiority over *No Domains* shows that learning contributions of individual domains in a fine-grained way is crucial for prediction quality; the aggregate volume of a startup's web presence is simply not all that matters. This answers **RQ4**.

6.2 Feature group contributions

Having established WBSSP's strength, we now proceed to measure the importance of each feature group for the best-performing final model (**RQ3**). Following [7], we define the *strength* of a feature f_i to be the expected squared output change of classifier c when f_i is removed, averaged over the trees in the ensemble:

$$
\begin{aligned}
&Str_c(f_i)\\
&= \sum_{t \in \mathrm{Trees(c)}} \mathbb{E}_x \left(c(x) - c_{\setminus f_i}(x) \right)^2\\
&= \sum_{t=1}^{T} \sum_{l=1}^{L_t} \left(c(t,l) - \frac{c(t,l)|L_{c,t}(l)| + c_{\setminus f_i}(t,l)|L_{c_{\setminus f_i},t}(l)|}{|L_{c,t}(l)| + |L_{c_{\setminus f_i},t}(l)|} \right)^2 |L_{c,t}(l)|\\
&\quad + \left(c_{\setminus f_i}(t,l) - \frac{c(t,l)|L_{c,t}(l)| + c_{\setminus f_i}(t,l)|L_{c_{\setminus f_i},t}(l)|}{|L_{c,t}(l)| + |L_{c_{\setminus f_i},t}(l)|} \right)^2 |L_{c_{\setminus f_i},t}(l)|.
\end{aligned}
$$

In the above formula, with a slight abuse of notation, we write $c_{\setminus f_i}$ for the classifier trained without feature f_i; T and L_t are the number of trees and leaves in tree t; $c(t,l)$ is the output of leaf l in tree t of classifier c; and $|L_{c,t}(l)|$ is the number of samples belonging to leaf l. For simplicity, the strength of a group of features F is then defined as the sum of corresponding feature strengths: $Str_c(F) = \sum_{f \in F} Str_c(f)$. Results of the analysis are shown in Fig. 4.

As in the previous section, *Investor*, *General* and *Mentions* features influence our model the most. Moreover, *Mentions* are the second strongest group of the four, which also confirms the second hypothesis of **RQ3**. An interesting observation is that *Team* is the weakest feature group; it provides empirical evidence for

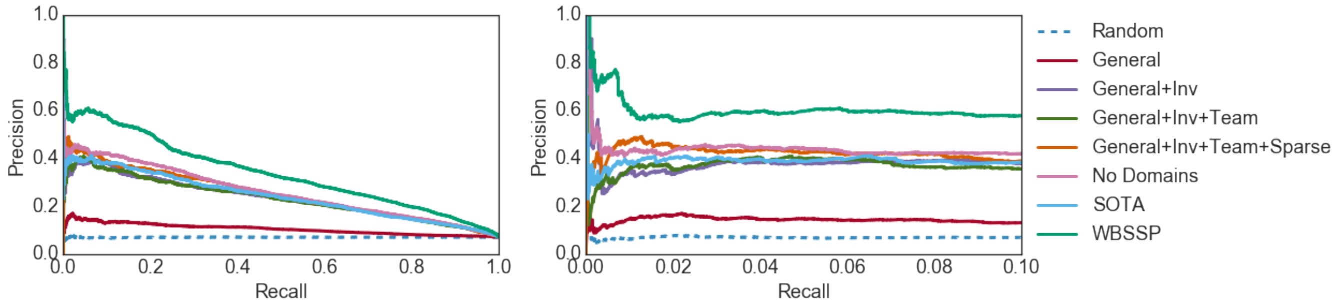


Figure 3: Precision-Recall (PR) curve for different feature groups and baselines. Left: full PR curve. Right: zooming in on the low-recall region, which is the most relevant from a real-world point of view. *Inv* stands for Investor features.

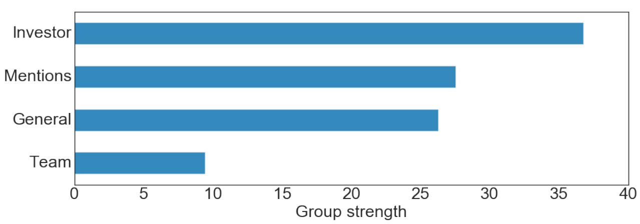


Figure 4: Feature strengths, as defined in Section 6.2, for different feature groups. Investor, general and mention features are considered most important by the model.

the findings of [16], which states that investors in a startup should generally place more weight on the business ("Horse") rather than on the founding team ("Jockey").

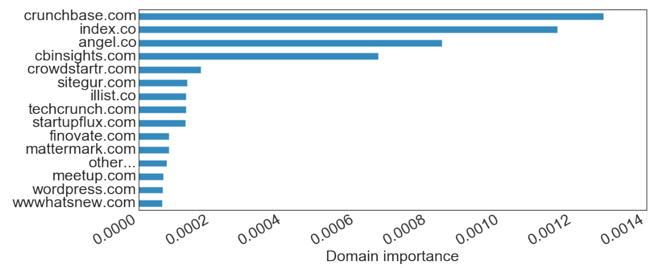
6.3 Individual domain contributions

In Section 6.1.2, we have established that learning from individual domains (*WBSSP*) yields a performance boost in comparison to only using aggregate mention amounts statistics (*No Domains*). Thus, as pointed out in our anwer to **RQ4**, it is important to understand which domains influenced our model's predictions the most. We consider the weights that were assigned to each domain by a LR that treats sparse factors; see Table 1, *links_domains_flag_total* features. We define a LR's feature importance as the fraction of total prediction variance contributed by that feature; in an online-trained model this is formalized as

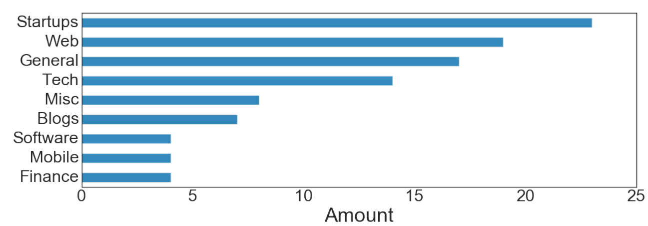
$$Imp(i) = \frac{\mathrm{Var}\left(\{x_i^{(t)} \cdot w_i^{(t)}\}_{t=1}^{|X|}\right)}{\mathrm{Var}\left(\{\sum_{j=1}^{|F|} x_j^{(t)} \cdot w_j^{(t)}\}_{t=1}^{|X|}\right)}, \tag{1}$$

where $x^{(t)} \in X$ are indexed by t in ascending temporal order, $w^{(t)}$ is the feature weight vector learned by step t, and F is the set of Logistic Regression features.

We show the top 15 domains in terms of importance (Eq. 1) in Fig. 5 (a); moreover, we manually classify the top 100 domains into 9 broad categories and show their relative populations in Fig. 5 (b). The categories are: startup and entrepreneurship-related resources, e.g., *venturebeat.com* or *owler.com* (*Startups*); news and articles on technology, e.g., *techrepublic.com* (*Tech*), finance, e.g., *forbes.com* (*Finance*), software, e.g., *github.com* (*Software*); mobile products and applications, e.g., *apple.com* (*Mobile*); blogs, e.g., *blogspot.ru* (*Blogs*); web-related resources and aggregators, e.g., *siterankd.com* (*Web*); all-around news or knowledge portals, e.g., *cnn.com* (*General*); and



(a)



(b)

Figure 5: The most important individual mentions sources: (a) importances of top 15 domains as identified by our model, (b) fractions of different types of domains from the top 100 domains ranked by importance score as defined in Eq. 1.

other types of resources (*Misc*). From Fig. 5 (a) it can be seen that the top important domains, indeed, are mostly significant entities in the startup and business world,[12] including *index.co*, *angel.co* (both startup-investor connecting social networks), *finovate.com* (major startup-related conference) etc. Fig. 5 (b), however, shows that the important domains are diverse and not limited to specialized startup-related resources: a large part of the top 100 consists of web-related resources, both broad and tech-specific news portals.

6.4 Scale importance analysis

Finally, we address the hypothesis posed in **RQ5**. We seek to check whether a startup's web presence signal actually adheres to the "wisdom of the crowd" intuition: we expect mentions to become an

[12]Note that *crunchbase.com* is the top ranked domain, which is to be expected, given that our sample is biased towards companies on Crunchbase. However, this domain is not the sole decisive contribution, since our quality metrics did not change significantly when retraining WBSSP without *crunchbase.com*.

increasingly stronger signal source as the number of gathered mentions increases, which is equivalent to the "crowd" getting larger. To that end, we trained WBSSP on data including only subsamples of the total available mentions; that is, each mention is independently included in the dataset with probability p. All other types of features were included without changes. We experiment with several values of p and report the ROC-AUC scores in Fig. 6.

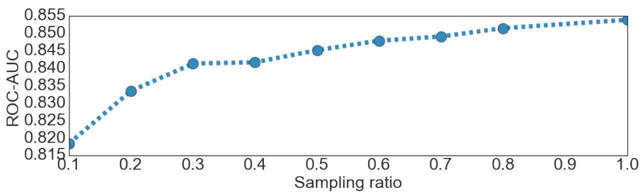


Figure 6: ROC-AUC values for different fractions of mentions included in the model.

In conclusion, we see that the quality behaves as expected when increasing the amount of aggregated mentions, providing evidence supporting the hypothesis posed in **(RQ5)**. These results suggest that improving the mentions mining process is a possible direction for future work; see Section 7.

7 CONCLUSIONS

In this paper, we addressed the problem of predicting the success of startup companies during their early stages of development. We utilized a rich and heterogeneous set of signals including data both from Crunchbase, the largest open-access startup database, and from a crawl of web-based open sources. We also developed a robust and diversified prediction pipeline, WBSSP, based on a combination of several machine learning models; conducting the largest experiments on the problem so far, we show that our method exceeds the current state-of-the-art by a large margin.

Besides building a predictive model, we contributed by providing a thorough analysis of this model and obtained results. Quite expectedly, structured company data such as category, investor data etc. is important for predictions. However, a significant finding of our work is the usefulness of taking a company's web presence in the form of mentions on different websites into account: while not being significant individually, these mentions, upon aggregation, form a representative picture of a company's perception by its target audience and significantly improve the quality of predictions.

Despite the fact that we have addressed various limitations of previous research into startup success prediction, our work highlights several opportunities for improvement. First, in addition to tracking only the sources of startup mentions, further work should also make use of the contents of the discovered mentioning pages, e.g., in the form of sentiment analysis. Second, as the experiments of Section 6.4 show, prediction quality does not saturate when the amount of incorporated mentions approaches our full dataset. Since our study was limited to using direct mentions in the form of links, further work may focus on discovering indirect mentions, for example, by company name, with the use of Named Entity Recognition techniques. Finally, we have only considered domain-level mentions, that is, different web pages or second-level domains within a broader domain were considered identical. While this is justified for small web resources, large domains such as, e.g., *reddit.com* or *forbes.com* comprise a huge number of sections on very diverse topics. Further distinguishing between these sections may provide us with a more fine-grained signal for predictive modeling.

REFERENCES

[1] R. Aggarwal and H. Singh. Differential influence of blogs across different stages of decision making: The case of venture capitalists. *MIS Q.*, 37(4):1093–1112, Dec. 2013.
[2] M. D.-K. Ayyagari and V. Asli Maksimovic. *Small vs. Young Firms across the World: Contribution to Employment, Job Creation, and Growth.* The World Bank, 2011.
[3] R. Biggadike. The risky business of diversification. *Harvard Business Review*, 57 (3):103–111, 1979.
[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.
[5] L. Bottou. Online algorithms and stochastic approximations. In D. Saad, editor, *Online Learning and Neural Networks*. Cambridge University Press, Cambridge, UK, 1998. revised, oct 2012.
[6] CatBoost. Gradient boosting on decision trees library. https://github.com/catboost, 2017.
[7] CatBoost. Regular feature importance. https://tech.yandex.com/catboost/doc/dg/concepts/fstr-docpage/, 2017.
[8] J. F. Coyle and G. D. Polsky. Acqui-hiring. *Duke Law Journal*, 63, 2012.
[9] A. Davila, G. Foster, and M. Gupta. Venture capital financing and the growth of startup firms. *Journal of Business Venturing*, 18(6):689–708, 2003.
[10] P. Gompers, W. Gornall, S. N. Kaplan, and I. A. Strebulaev. How do venture capitalists make decisions? Techn. report, Nat. Bureau of Econ. Research, 2016.
[11] C. E. Halabí and R. N. Lussier. A model for predicting small firm performance: Increasing the probability of entrepreneurial success in chile. *Journal of Small Business and Enterprise Development*, 21(1):4–25, 2014.
[12] D. K. Hsu, J. M. Haynie, S. A. Simmons, and A. McKelvie. What matters, matters differently: a conjoint analysis of the decision policies of angel and venture capital investors. *Venture Capital*, 16(1):1–25, 2014.
[13] A. G. Huang and K. Mamo. Do analysts read the news? In *Canadian Academic Accounting Association (CAAA) Annual Conference*, May 2016.
[14] A. Hyytinen, M. Pajarinen, and P. Rouvinen. Does innovativeness reduce startup survival rates? *Journal of Business Venturing*, 30(4):564 – 581, 2015.
[15] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015.
[16] S. N. Kaplan, B. A. Sensoy, and P. Strömberg. Should investors bet on the jockey or the horse? Evidence from the evolution of firms from early business plans to public companies. *The Journal of Finance*, 64(1):75–115, 2009.
[17] W. R. Kerr, J. Lerner, and A. Schoar. The consequences of entrepreneurial finance: Evidence from angel financings. *The Review of Financial Studies*, 27(1):20–55, 2011.
[18] S. Kortum and J. Lerner. Does venture capital spur innovation? In *Entrepreneurial inputs and outcomes: New studies of entrepreneurship in the United States*, pages 1–44. Emerald Group Publishing Limited, 2001.
[19] Y. E. Liang and S.-T. D. Yuan. Investors are social animals: Predicting investor behavior using social network features via supervised learning approach. In *Eleventh Workshop on Mining and Learning with Graphs (MLG 2013)*. ACM, 2013.
[20] R. N. Lussier. A business success versus failure prediction model for service industries. *Journal of Business and Entrepreneurship*, 8(2):23, 1996.
[21] C. M. Mason and T. Botelho. Comparing the initial screening of investment opportunities by business angel group gatekeepers and individual angels. In *2017 Emerging Trends in Entrepreneurial Finance Conference*, March 2017.
[22] A. L. Maxwell, S. A. Jeffrey, and M. Lévesque. Business angel early stage decision making. *Journal of Business Venturing*, 26(2):212 – 225, 2011.
[23] S. B. Moeller, F. P. Schlingemann, and R. M. Stulz. Wealth destruction on a massive scale? a study of acquiring-firm returns in the recent merger wave. *The Journal of Finance*, 60(2):757–782, 2005.
[24] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, pages 807–814, 2010.
[25] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
[26] T. Stone, W. Zhang, and X. Zhao. An empirical study of top-n recommendation for venture finance. In *CIKM*, pages 1865–1868. ACM, 2013.
[27] J. Surowiecki, M. P. Silverman, et al. The wisdom of crowds. *American Journal of Physics*, 75(2):190–192, 2007.
[28] A. Wong, M. Bhatia, and Z. Freeman. Angel finance: the other venture capital. *Strategic Change*, 18(7-8):221–230, 2009.
[29] G. Xiang, Z. Zheng, M. Wen, J. I. Hong, C. P. Rosé, and C. Liu. A supervised approach to predict company acquisition with factual and topic features using profiles and news articles on techcrunch. In *ICWSM*. AAAI, June 2012.
[30] C. Zhang, E. Chan, and A. Abdulhamid. Link prediction in bipartite venture capital investment networks. Stanford University, 2015.
[31] Q. Zhang, T. Ye, M. Essaidi, S. Agarwal, V. Liu, and B. T. Loo. Predicting startup crowdfunding success through longitudinal social engagement analysis. In *CIKM*, pages 1937–1946. ACM, 2017.
[32] H. Zhong, C. Liu, J. Zhong, and H. Xiong. Which startup to invest in: a personalized portfolio strategy. *Annals of Operations Research*, pages 1–22, 2016.