Building Trusted Startup Teams From LinkedIn Attributes: A Higher Order Probabilistic Analysis

Georgios Drakopoulos
Ionian University
c16drak@ionio.gr

Eleana Kafeza
Zayed University
eleana.kafeza@zu.ac.ae

Phivos Mylonas
Ionian University
fmylonas@ionio.gr

Haseena al Katheeri
Zayed University
haseena.alaktheeri@zu.ac.ae

***Abstract*—Startups arguably contribute to the current business landscape by developing innovative products and services. The discovery of business partners and employees with a specific background which can be verified stands out repeatedly as a prime obstacle. LinkedIn is a popular platform where professional milestones, endorsements, recommendations, and skills are posted. A graph search algorithm with a BFS and a DFS strategy for seeking trusted candidates in LinkedIn is proposed. Both strategies rely on a metric for assessing the trustworthiness of an account according to LinkedIn attributes. Also, a stochastic vertex selection mechanism reminiscent of preferential attachment guides search. Both strategies were verified against a large segment of the vivid startup ecosystem of Patras, Hellas. A higher order probabilistic analysis suggests that BFS is more suitable. Findings also imply that emphasis should be given to local networking events, peer interaction, and to tasks allowing verifiable credit for the respective work.**

***Index Terms*—trust, linked data, graph mining, probabilistic analysis, higher order statistics, multilayer graphs, LinkedIn API**

I. Introduction

The ongoing worldwide digital transformation relies heavily on startups for paving the way for ground-breaking innovation. To this end, incubators, accelerators, and other stakeholders dedicate considerable effort in startup formation. Among the most important obstacles, especially during the early formulation stage, is the discovery of trusted business partners or employees with a specific skillset [1], corroborating the position that human capital, including trust, is perhaps the most critical factor in the innovation process [2]. For our purposes the following definition will be used:

Definition 1 (Trusted candidate): A candidate is considered to be trusted if and only if his/her accomplishments can be verified by available online resources.

Definition 1 raises among others the question of how such a candidate with a certain skillset can be verified in platforms like LinkedIn. A wholly owned subsidiary of Microsoft for 26.2 billion dollars[1] since December 2016, LinkedIn as of May 2020 has almost 700 million accounts from 150 countries around the globe. Although it is difficult to be directly evaluated, in LinkedIn trust can manifest itself in ways which can be translated to metrics. In fact, LinkedIn has taken a step towards that direction by introducing skill assessments in September 2019[2]. Thus, navigating the LinkedIn graph and discovering trusted candidates based on its attributes constitutes the principal motivation of this work.

The primary research contribution of this conference paper is a LinkedIn graph search algorithm with a breadth first search (BFS) and a depth fist search (DFS) mode. Both variants, programmed in Python 3.8, rely on a stochastic vertex selection mechanism, akin to preferential attachment [3], utilizing attributes retrieved by the LinkedIn application interface (API). They have been tested on a graph representing a major part of the growing startup ecosystem of Patras, Hellas with encouraging results. Probabilistic analysis of the number of steps and the number of steps to the first trusted candidate suggests that the BFS mode outperforms the DFS one.

The remaining of this work is structured as follows. In section II the scientific literature regarding graph mining, trust, and higher order statistics is briefly reviewed. The proposed algorithms and their basic parameters are given in section III. Section IV contains the test dataset, the verification process, the probabilistic analysis, and recommendations based on the latter. Finally, the main findings as well as a discussion about possible research directions are given in section V. Technical acronyms are explained the first time they are encountered in the text. To avoid confusion, Web site retrieval dates are given in US military style. Random variables are represented by capital calligraphic letters. In a function definition parameters are placed after its arguments following a semicolon. Finally table I summarizes the notation of this work.

II. Previous Work

LinkedIn has been the focus of various studies such as [4]. From a technical perspective, its underlying data infrastructure [5] and how it handles big data [6] have been analyzed. The way LinkedIn profiles are shaped across professions is the focus of [7], whereas [8] examines their role in the informatics and communications technologies (ICT) sector. Similarities and differences between profiles in LinkedIn and social media such as Twitter and Facebook are explored in [9] and in [10]. Moreover, [11] explains how LinkedIn self-descriptions

[1]https://news.microsoft.com/announcement/microsoft-buys-linkedin (retrieved 4.jul.2020)

[2]https://blog.linkedin.com/2019/september/17/announcing-skill-assessments-to-help-you-showcase-your-skills (retrieved 4.jul.2020)

TABLE I
NOTATION OF THIS WORK.

Symbol	Meaning
$\triangleq$	Definition or equality by definition
$\{s_1, \ldots, s_n\}$	Set with elements $s_1, \ldots, s_n$
$\lvert S\rvert$ or $\lvert\{s_1, \ldots, s_n\}\rvert$	Set cardinality
$\tau(S_1, S_2)$	Tanimoto set similarity coefficient
$\nu(T, V; \alpha_0)$	Asymmetric Tversky set similarity index
$\Gamma(v)$	Neighbourhood of vertex v
$S_1 \setminus S_2$	Asymmetric set difference of S_1 minus S_2
$\mathrm{E}[\mathcal{X}]$	Expected value of random variable $\mathcal{X}$
$\mathrm{Var}[\mathcal{X}]$	Variance of random variable $\mathcal{X}$
$\bar{\mu}_3[\mathcal{X}]$	Skewness of random variable $\mathcal{X}$
$\bar{\mu}_4[\mathcal{X}]$	Kurtosis of random variable $\mathcal{X}$
$\langle p \,\|\| \, q\rangle$	Kullback-Leibler divergence between p and q

intended for friends or employers differentiate from each other. Also, the role of inaccurate LinkedIn resumes and how they can be detected are investigated in [12].

Higher order statistics focuses on the study of moments and cumulants beyond the first and second order ones [13]. Applications include signal processing [14], nonlinear system identification [15], biomedical image analysis [16] and electroencephalography (EEG) [17] processing, and time series sensitivity analysis [18]. Moreover, cumulants have a close connection to tensor stack networks (TSN) [19].

Graph mining, as its name suggests, is the field of extracting knowledge from graphs [20] and also a major driver behind mining patterns in massive, linked, and (semi)structured datasets [21]. Structural patterns are related to combinatorial properties such as triangles [22], cyclic decompositions [23], low rank approximations [24], and community discovery [25] [26]. Nonlinear diffusion for the latter is proposed in [27] and partitioning of skewed graphs in [28]. Functional patterns rely heavily on the type of operations on graphs such as spatio-linguistic tweet analysis [29], attention models [30], digital influence [31], and anomaly discovery [32].

III. Algorithmic Approach

A. Patras Startup Ecosystem

At this point the source of the test dataset is described. Patras, the third largest Hellenic city is located in a strategic position in Patraikos bay and is the primary commercial and cultural gate to Central and Western Europe as well as North Africa. Becasue of this, it has been continuously since late renaissance a cultural reference and the home of scholars, artisans, engineers, and one of the oldest Greek universities.

Currently, the startup ecosystem of Patras is considered one of the most thriving in the country with incubators such as Orange Grove[3], NGOs supporting innovation like Mindspace[4], a science park[5], a local IEEE student branch including a biomed chapter, and three universities. During the past three years Patras has been visited by the US[6], French, and Dutch ambssadors. Moreover, there have been visits from high ranking policymakers such as trade and cultural attachés and state officials. The ecosystem has been recently mentioned from US Embassy because of a high profile startup buyout[7].

At this point it should be emphasized that the proposed technique can be applied to other enterprise entities. However, the case of startups has been selected for the following reasons:

- Larger companies have sophisticated human resources (HR) tools and recruitment methodologies for determining candidate trustworthiness based on definition 1.
- Locating trusted candidates has been reported as a prime problem for startups, especially during their early stages.

B. Graph Building And Skill Mapping

LinkedIn profiles contain a plethora of professional infromation including background, skills, previous positions, and professional career milestones. The proposed algorithms rely on endorsements and self-descriptions to discover skills as well as on recommendations to measure candidate trustworthiness. The values of variables discussed here are given in table IV.

The test dataset is a LinkedIn subgraph consisting of:

- v_s vertices, with v_k^s for the k-th startup page.
- v_c vertices, with v_j^c for the j-th candidate profile.
- u edges representing actual LinkedIn connections.

Through RAKE algorithm [33], an alternative to tf-idf and TextRank, for each startup the most frequent technologies were identified. Then the top N_c most common were selected along with the number of startups listing them as core competencies, resulting in table II. For the purposes of this text this is the universe S_0 of all possible tehnologies. Therefore, the k-th startup $1 \leq k \leq v_s$ has its own technology set $S_k \subseteq S_0$ with elements drawn from S_0.

TABLE II
FREQUENT STARTUP TECHNOLOGIES.

Technology	Startups	Technology	Startups
Data mining	36	Android	11
Blockchain	22	IoT	11
NoSQL databases	16	GPU computing	7
Relational databases	16	NLP	7
Social media	14	GIS	2

Once the candidate skills were extracted, again through RAKE, they were uniquely mapped to the technologies of table II using the computer science ontology proposed in [34]. The number of skills as well as the most frequent skills for each technology are given in table III. Thus, the j-th candidate $1 \leq j \leq v_c$ has its own skill set $C_j \subseteq S_0$.

Concerning trust, it can be derived in two ways:

- Explicitly, by a mention that the candidate can be trusted or that she/he has a certain skill.
- Implicitly when a certain skill is highly endorsed.

Notice that both ways require assessments from LinkedIn members. This stems from the fact that trust is a human trait and as such it is better left to humans than to algorithms.

[3] www.orangegrove.eu

[4] www.mindspace.gr

[5] www.psp.org.gr

[6] https://gr.usembassy.gov/ambassador-pyatt-visits-patras (retrieved 4.jul.2020)

[7] https://gr.usembassy.gov/ambassador-pyatts-video-remarks-at-the-8th-regional-growth-conference-in-patras-july-3-2020 (retrieved 4.jul.2020)

Regarding the explicitly stated trust, suppose that the j-th candidate has r_j recommendations. The i-th of them may mention that the respective candidate is trustworthy or that he/she has one or more skills relevant to the technologies of S_0. In the former case, all skills of C_j are marked as trusted and a counter q_i is set to $|C_j|$, whereas in the latter q_i counts only those skills mentioned. Thus, the explicit trust π_j^e is given as in (1). The keywords denoting trust were taken from the online dictionary of the Oxford University Press[8].

$$\pi_j^e \triangleq \frac{1}{r_j}\sum_{i=1}^{r_j}\frac{q_i}{|C_j|} \;=\; \frac{1}{r_j|C_j|}\sum_{i=1}^{r_j} q_i \tag{1}$$

The implicit trust can be evaluated as follows. Assume that the i-th skill of the j-th candidate has n_i endorsements whose sum equals t_j, ignoring endorsements to skills which were not mapped to the elements of S_0. The number of implictly trusted skils c_j is that when the normalized sum of sorted n_i in descending order first exceeds a threshold η_0:

$$\frac{\sum_{i=1}^{c_j} n_i}{\sum_{l=1}^{|C_j|} n_l} \;=\; \frac{\sum_{i=1}^{c_j} n_i}{t_j} \;\geq\; \eta_0, \qquad 0 \;<\; \eta_0 \;\leq\; 1 \tag{2}$$

Notice that equation (2) rewards candidates with balanced profiles. The implicit trust π_j^m is given in equation (3):

$$\pi_j^m \triangleq \frac{c_j}{|C_j|} \tag{3}$$

Finally, the trust π_j for the j-th candidate is the weighted sum of (4). ρ_0 is a hyperparameter indicating the relative importance of the explicit trust compared to the implicit one.

$$\pi_j \triangleq \frac{1}{1+\rho_0}\pi_j^m + \frac{\rho_0}{1+\rho_0}\pi_j^e \tag{4}$$

C. Stochastic Graph Search

The stochastic graph search comes in two flavors, a BFS and a DFS one described in algorithms 1 and 2 respectively. Both select at each step the next vertex to visit on a stochastic basis. To simplify analysis, it will be assumed that the search only discovers a v_j^c. The same steps apply when a v_k^s is found.

Here it should be noted that there are only two differences of BFS and DFS searches from their textbook counterparts:

- The next vertex is selected based on a probabilistic mechanism depending on local skillset similarities.
- Once a vertex is visited, a metric determines how well a candidate matches with a given startup.

Thus, here only these two differences will be explained.

Concerning the first difference, assume that the search is currently at $v_{j'}^c$. Then, for each $v_j^c \in \Gamma\left(v_{j'}^c\right)$ and also for each $v_i^c \in \Gamma\left(v_j^c\right)$ the Tanimoto coefficient is applied as in (5):

$$\tau\left(C_{j'}, C_j\right) \triangleq \frac{|C_{j'} \cap C_j|}{|C_{j'} \cup C_j|} \;=\; \frac{|C_{j'} \cap C_j|}{|C_{j'}| + |C_j| - |C_{j'} \cap C_j|} \tag{5}$$

The second form of (5) is more efficient for large sets. Also, set cardinality estimators such as [35] or [36] can be used.

[8]https://dictonary.cambridge.org (retrieved 4.jul.2020)

Then, each admissible vertex $v_j^c \in \Gamma\left(v_{j'}^c\right)$ is assigned a probability as in (6). The latter can be thought of a weighted version of the preferential attachment mechanism.

$$\text{prob}\left\{v_{j'}^c \to v_j^c\right\} \;\propto\; \frac{\tau\left(C_{j'}, C_j\right)}{\sum_{v_i^c}\tau\left(C_i, C_j\right)}, \quad v_i^c \in \Gamma\left(v_j^c\right) \tag{6}$$

Regarding the second difference, assume that search is currently at v_j^c and that v_k^s, listing its core competencies in S_k, is the starting point. The search finds only trusted candidates for that startup, as different startups are located in different parts of the graph, meaning that local connectivity steps will inevitably vary. Once at v_j^c, its skillset compatibility is assessed through the asymmetric Tversky index [37]. In contrast to (5), S_k is the template against which C_j is compared. As S_k and C_j are not equivalent semantically, the Tversky index is a logical choice. On the other hand, there is no need to distinguish between skillsets of candidates.

$$\nu\left(S_k, C_j; \alpha_0\right) \triangleq \frac{|S_k \cap C_j|}{|S_k \cap C_j| + \alpha_0|S_k \setminus C_j| + (1-\alpha_0)|C_j \setminus S_k|} \tag{7}$$

In (7) the hyperparameter $0 \leq \alpha_0 \leq 1$ specifies the relative importance of the second term of the denominator compared to the third one. Assuming a ratio of β_0, then:

$$\frac{\alpha_0}{1-\alpha_0} \;=\; \beta_0 \;\Leftrightarrow\; \alpha_0 \;=\; \frac{\beta_0}{1+\beta_0} \tag{8}$$

From equation (8) it follows that as β_0 grows, then α_0 tends asymptotically to one. Moreover, the rate of convergence to that asymptotic limit is progressively getting slower since:

$$\frac{\partial\alpha_0}{\partial\beta_0} \;=\; \frac{1}{\left(1+\beta_0\right)^2}\bigg|_{\beta_0 \to +\infty} \to 0 \tag{9}$$

Finally, the metric combining skillset combatibility and trust is given by their weighted harmonic mean in equation (10). The harmonic mean is less sensitive to outliers and can handle zero values in the denominator. As was the case with both previous hyperparameters, ρ_1 determines the relative weight between skillset compatibility and trust.

$$J \triangleq \frac{1+\rho_1}{\dfrac{1}{\nu\left(S_k, C_j\right)} + \dfrac{\rho_1}{\pi_j}} \;=\; \frac{(1+\rho_1)\pi_j\nu\left(S_k, C_j\right)}{\pi_j + \rho_1\nu\left(S_k, C_j\right)} \tag{10}$$

Once J is computed, a number of options is available:

- Keep only v_j^c with a J score over a threshold.
- Keep a number of v_j^c scoring above a threshold.
- Keep a window v_j^c with highest J scores.
- Keep all v_j^c and rank them after search is over.

In practice, for large graphs there can be a maximum number of steps. This should not be confused with the above options.

Algorithm 1 is a BFS implementation based on a queue, namely a first-in first-out (FIFO) array. The latter supports two primary operations, namely *queue* where a vertex is placed in the queue and *dequeue* where a vertex is extracted from it.

Algorithm 2 is an implementation of a stack, namely a last-in, first out (LIFO) array. Like a queue, a stack supports two

TABLE III
SKILL MAPPING TO STARTUP TECHNOLOGIES.

Technology	Frequent Related Skills	Skills
Data mining	Spark, Hive, R, MATLAB, data science, pattern recognition, statistics, mathematical modeling	77
Blockchain	Smart contracts, electronic contracts, Solidity	12
NoSQL databases	BASE, MongoDB, Neo4j, Cassandra, HBase, column store	27
Relational databases	SQL, PostgreSQL, tabular data, OLAP cube	39
Social media	Twitter, Facebook, social media analytics	33
Android	Mobile, mobile development, smart app development	4
IoT	internet of devices, embedded software, pervasive computing	25
GPU computing	CUDA, NVIDIA, OpenCL, compute kernel	12
NLP	natural language processing, regular expressions, stemming, lemmatization	8
GIS	spatial data analysis, geographical systems, geocoding	7

Algorithm 1 Trusted Candidate Discovery - BFS version

Require: Hyperparameters α_0, ρ_0, ρ_1
Ensure: Discover a set of trusted candidates
start from $v^s_{k'}$, **mark** it, **enqueue** $v^s_{k'}$
while queue is **not** empty **do**
  dequeue $v^c_{j'}$
  for all $v^c_j \in \Gamma\left(v^c_{j'}\right)$ **do**
    compute the next vertex as in (5)
    if v^c_j is **not** marked **then**
      mark v^c_j, **compute** J as in (10), **enqueue** v^c_j
    end if
  end for
end while

operations, namely *push* and *pop*. The former inserts a vertex in the stack, whereas the latter extracts the last vertex inserted.

Algorithm 2 Trusted Candidate Discovery - DFS version

Require: Hyperparameters α_0, ρ_0, ρ_1
Ensure: Discover a set of trusted candidates
start from $v^s_{k'}$, **push** $v^s_{k'}$
while stack is **not** empty **do**
  pop v^c_j
  if vertex v^c_j is **not** marked **then**
    compute J as in (10), **mark** v^c_j
    for all $v^c_i \in \Gamma\left(v^c_j\right)$ **do**
      compute the next vertex as in (5), **push** v^c_i
    end for
  end if
end while

IV. RESULTS

A. Test Dataset

LinkedIn offers a publicly available API with extensive documentation[9]. It is currently in version 2.0, which is mandatory since March 2019 whereas version 1.0 is bound to be designated as deprecated, and relies on OAuth and TLS 1.2 for secure communication. This API includes functionality for managing group memberships and postings[10] among others.

There are several limitations to collecting data. Every UTC midnight the number of API calls of an application or of a single member per application are reset[11]. Additionally, for security reasons connections beyond the fourth degree of the application's owner account cannot be accessed. To overcome the above limitations, the data have been collected during April 2020 with a steady rate since bursty or abnormal activity patterns have been reported to be banned[12].

Table IV contains a synopsis of the dataset properties and the values of any variables mentioned in our analysis.

TABLE IV
TEST DATASET SYNOPSIS.

Property name	Numerical value
Vertices v_s (startups)	47
Vertices v_c (candidates)	6391
Edges u (max number of edges)	1512388 (20720703)
Triangles (max number of triangles)	314003 ($1.48e+10$)
Squares (max number of squares)	109863 ($1.78e+13$)
Connected components	1
Density δ_0 / Log-density δ'_0	234.9518 / 1.6224
Completeness δ_1 / Log-completeness δ'_1	0.0729 / 1.7617
Graph diameter L_0	10
Fraction of vertices with distance 6	15.66%
Fraction of vertices with distance 7	12.33%
Fraction of vertices with distance 8	7.81%
Fraction of vertices with distance 9	4.68%
Fraction of vertices with distance $L_0 = 10$	1.15%
Total number of technologies N_s	10
Mean of skills per profile (after mapping)	7.13
Hyperparameter α_0	2/3
Hyperparameter ρ_0	1
Hyperparameter ρ_1	Varies (see table V)
Number of search runs R_0	100
Maximum number of vertex visits V_0	2000
Variable τ for estimating Chernoff bounds	Uniform in $[1, v_s]$
Runs of Chernoff bounds	1000

From table IV it can be deduced that the test graph is very connected. This can be attributed to the following reasons:

[9]https://www.linkedin.com/developers (retrieved 4.jul.2020)

[10]https://docs.microsoft.com/en-us/linkedin/compliance/integrations/groups/group-memberships (retrieved 4.jul.2020)

[11]https://docs.microsoft.com/en-us/linkedin/shared/api-guide/concepts/rate-limits?context=linkedin/context (retrieved 4.jul.2020)

[12]https://www.linkedin.com/help/linkedin/82934/account-content-restricted-or-removed (retrieved 4.jul.2020)

- The average degree (see equation (11)) is high. This can be explained as LinkedIn is designed for networking.
- The majority of pairwise vertex distances is low.
- The number of triangles and squares is high. Although much lower than the theoretical maximum, the average number of triangles and connections per candidate are very close, indicating considerable local density.

Triangles are unique in the sense that they are third order patterns for both vertices and edges and also cliques of size three. On the other hand, squares, and higher order shapes for that matter, are fourth order patterns for vertices and edges but do not constitute a clique of size four which is a fourth order for vertices but a sixth order one for edges.

Density δ_0 is the ratio of the total number of edges to the total number of edges as shown in equation (11), which is an approximation to the average vertex degree.

$$\delta_0 \triangleq \frac{v_s + v_c}{u} \approx \frac{v_c}{u} \tag{11}$$

Log-density δ_0' is another metric of graph connectivity, defined as the ratio of the order of magnitude of the number of edges to the order of magnitude of the number of vertices as shown in equation (12):

$$\delta_0' \triangleq \frac{\ln(v_s + v_c)}{\ln u} \approx \frac{\ln v_c}{\ln u} \tag{12}$$

Completeness δ_1 measures how many edges a given graph has compared to the total number of edges a complete graph with the same number of vertices as seen in equation (13):

$$\delta_1 \triangleq \frac{u}{\binom{v_c+v_s}{2}} \approx \frac{u}{\binom{v_c}{2}} \approx \frac{2}{v_c \delta_0} \tag{13}$$

Log-completeness δ_1' defined in equation (14) follows the same line of reasoning as log-density:

$$\delta_1' \triangleq \frac{\ln u}{\ln \binom{v_c+v_s}{2}} \approx \frac{\ln u}{\ln \binom{v_c}{2}} \approx \frac{1}{2\delta_0'} \tag{14}$$

The exact values of δ_0, δ_0', δ_1, and δ_1' have been computed. The above metrics besides assessing graph connectivity are also inherently tied to the evaluation of the total graph worth in structural terms similarly to Metcalf's law, which is historically among the earliest such efforts [38].

Since the proposed algorithms are stochastic, for each startup v_k^s both searches were each run R_0 times and the mean values were kept. The upper limit of the vertices which could be visited was V_0, however the actual limit for each run was much lower as will be discussed in the next subsection.

B. Candidate Trust Verification

To evaluate the ability of the proposed algorithms we rely on the observation that startups trust their current employees. Therefore, the accuracy shall be assessed in terms of the fraction of existing employees found. Specifically, if v_k^s has e_k employees, then both algorithms run until they $3e_k + 1$ candidates are found and then accuracy is computed. This is much lower than V_0, which acts as a failsafe mechanism.

Accuracy can be computed in two similar but not identical ways. The first is I_o of equation (15) which is defined as the ratio of the sum of the employees found e_k' for each v_k^s to the sum of the actual number of startup employees in the dataset:

$$I_o \triangleq \frac{\sum_{k=1}^{v_s} e_k'}{\sum_{k=1}^{v_s} e_k} \tag{15}$$

However, it can yield a somewhat loose bound since it can absorb a few low scores in the overall sum in the numerator.

An alternative yielding tighter bounds and allowing a probabilistic analysis is the accuracy indicator I_a defined as the average ratio of e_k' to e_k as shown in equation (16):

$$I_a \triangleq \frac{1}{v_s} \sum_{k=1}^{v_s} \frac{e_k'}{e_k} \approx \mathrm{E}\left[\frac{e_k'}{e_k}\right] = \mathrm{E}[\mathcal{E}] \tag{16}$$

This can be considered as the sample mean approximation of the stochastic mean of the random variable (r.v.) $\mathcal{E}$ containing true accuracy with each ratio being an observation of it. The stochastic variance of $\mathcal{E}$ can be estimated using the the sample variance, under mild conditions of ergodicity:

$$\mathrm{Var}[\mathcal{E}] \approx \sigma_a^2 \triangleq \frac{1}{v_s - 1} \sum_{k=1}^{v_s} \left(\frac{e_k'}{e_k} - I_a\right)^2 \tag{17}$$

Variance indicates whether there is a considerable concentration of $\mathcal{E}$ around I_a, acting as a reliability metric for the latter. A large concentration around I_a would be desirable, implying the graph search algorithms achieve satisfactory scores consistently. On the other hand, a high value of $\mathrm{Var}[\mathcal{E}]$ means that $\mathcal{E}$ fluctuates, revealing an erratic scoring performance.

TABLE V
ACCURACY AND VARIANCE FOR BFS AND DFS STRATEGIES.

ρ_1	I_a (**B/D**)	σ_a^2 (**B/D**)	I_o (**B/D**)
0.1	0.8621 / 0.8617	0.1844 / 0.1932	0.8133 / 0.7902
0.2	0.8680 / 0.8612	0.1803 / 0.2184	0.8134 / 0.7811
0.5	0.8739 / 0.8696	0.1599 / 0.1732	0.8142 / 0.8024
1	0.8377 / 0.8354	0.1809 / 0.2033	0.7889 / 0.7422
2	0.8146 / 0.8045	0.2216 / 0.2453	0.7785 / 0.7316
5	0.7803 / 0.7734	0.2567 / 0.2630	0.7533 / 0.7199
10	0.7723 / 0.7698	0.2633 / 0.2811	0.7312 / 0.7127

Values for I_a, σ_a^2, and I_o in relation to the hyperparameter ρ_1 of equation (10) are shown in table V. Recall that the relative weight of trust is inversely proportional to ρ_1. Observe that I_a is high whereas σ_a^2 is low, indicating a high confidence. I_o is is roughly the same with I_a. It is interesting that I_a is high for most values of ρ_1, indicating a balance in the way the matching metric works. In fact, the best values of I_a come when ρ_1 equals 0.5. However, when ρ_1 drops, then so does I_a, indicating that both algorithms discover trusted candidates. Also, BFS performs better than DFS in every case.

C. Total Number Of Steps Distribution

Another performance metric for algorithms 1 and 2 is their total number of steps, expressed in edge crossings. For the purposes of our analysis, let $\mathcal{H}^B$ and $\mathcal{H}^D$ denote the r.vs counting them for the BFS and the DFS strategy respectively.

Since the analysis will the same irrespective of the search strategy, let also $\mathcal{H}$ denote a generic r.v. standing in for both.

The first step is to determine the mean $\mathrm{E}[\mathcal{H}]$ and variance $\mathrm{Var}[\mathcal{H}]$. As in the previous subsection, they will be respectively approximated by their sample counterparts I_h and σ_h^2. Notice that a low value of both denotes not only that realistic expectations for finding trusted candidates can be had from the local startup ecosystem, but they are statistically valid.

Given I_h and σ_h^2 from a modeling perspective a first thought is to create a Gaussian model for the set of scores for $\mathcal{H}$:

$$f\left(x; I_h, \sigma_h^2\right) \triangleq \frac{1}{\sigma_h\sqrt{2\pi}} \exp\left(-\frac{\left(x - I_h\right)^2}{\sigma_h^2}\right) \tag{18}$$

Equation (18) is appealing for the following reasons:

- It has the maximum differential entropy among all distributions with the same expected value and variance [39].
- Its conjugate distribution is also the Gaussian, rendering the computation of a Bayesian estimator easy [40].

However, the results of normality tests taken from [41] argue against this hypothesis. To avoid cluttering, table VII has results only for the ρ_1 for the best I_h. The situation is the same for the rest of the values. Given that only the Gaussian distribution can be completely described by its mean and variance, a higher order analysis is therefore necessary.

At this point it is worth asking whether the number of steps can be bound. One answer comes from Markov inequality of equation (19), which works only for positive r.vs and requires only knowledge of the mean giving a first order bound:

$$\mathrm{prob}\left\{\mathcal{H} \geq \lambda_0\right\} \leq \frac{I_h}{\lambda_0}, \qquad \lambda_0 > 0 \tag{19}$$

A second order and also a power law bound can be obtained by the Chebyshev inequality of equation (20):

$$\mathrm{prob}\left\{\left|\mathcal{H} - I_h\right| \geq \lambda_1\sigma_h\right\} \leq \frac{1}{\lambda_1^2}, \qquad \lambda_1 > 0 \tag{20}$$

Chernoff bounds (21) are based on Markov inequality with the added assumption that a sum r.v. is composed of indepedent r.vs. They take advantage of the fact that $\mathcal{H}$ is a sum of v_s independent r.vs to obtain exponential bounds:

$$\mathrm{prob}\left\{\mathcal{H} \geq \lambda_2\right\} \leq \min_{\tau>0}\left[\exp\left(-\tau\lambda_2\right)\mathrm{E}\left[\prod_{l=1}^{v_s}\exp\left(\tau\mathcal{H}_l\right)\right]\right] \tag{21}$$

Figure 1 shows the bounds obtained by the three inequalities. The Chenroff bound is always lower, whereas BFS yields lower bounds than DFS because of the lower I_h and σ_h^2.

The skewness $\bar{\mu}_3[\mathcal{H}]$ of $\mathcal{H}$ is its third central normalized moment as shown in equation (22):

$$\bar{\mu}_3[\mathcal{H}] \triangleq \mathrm{E}\left[\left(\frac{\mathcal{H} - \mathrm{E}[\mathcal{H}]}{\sqrt{\mathrm{Var}[\mathcal{H}]}}\right)^3\right] = \frac{\kappa_3}{\kappa_2^{3/2}} \tag{22}$$

When $\bar{\mu}_3[\mathcal{H}]$ is finite its sign is an indication of the shape of the distribution of $\mathcal{H}$, if the latter is unimodal. Specifically:

- When $\bar{\mu}_3[\mathcal{H}]$ is zero, then the distribution is symmetric.

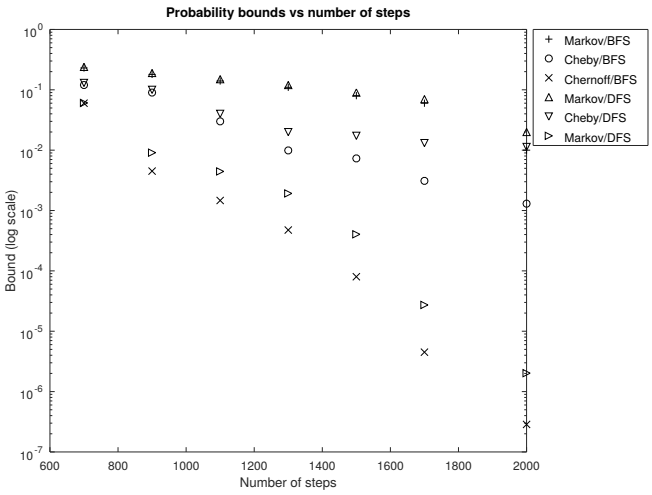

Fig. 1. Step Bounds For BFS And DFS Strategies ($\rho_1 = 0.5$).

- When $\bar{\mu}_3[\mathcal{H}]$ is positive, then the left tail is heavier.
- When $\bar{\mu}_3[\mathcal{H}]$ is negative, then the right tail is heavier.

In equation (22) the n-th cumulant κ_n is defined as the n-th coeccifient of the Taylor expansion of:

$$K(y) \triangleq \ln \mathrm{E}\left[e^{y\mathcal{H}}\right] = \sum_{n=0}^{+\infty}\kappa_n\frac{y^n}{n!} \tag{23}$$

Thus, the cumulant κ_n can be computed as in equation (24), on the condition that the series in (23) converges:

$$\kappa_n \triangleq \left.\frac{\partial^n K(y)}{\partial y^n}\right|_{y=0} \tag{24}$$

The tail weight of the distribution of $\mathcal{H}$ can be assesed by its kurtosis, defined as in equation (25):

$$\bar{\mu}_4[\mathcal{H}] \triangleq \mathrm{E}\left[\left(\frac{\mathcal{H} - \mathrm{E}[\mathcal{H}]}{\sqrt{\mathrm{Var}[\mathcal{H}]}}\right)^4\right] = \frac{\mathrm{E}\left[\left(\mathcal{H} - \mathrm{E}[\mathcal{H}]\right)^4\right]}{\mathrm{Var}[\mathcal{H}]^2} \tag{25}$$

Table VI shows that BFS yields systematically a lower number of total steps with higher confidence. Moreover, the skewness and kurtosis corroborate that the bulk of the BFS distribution is concentrated around a lower number of steps. Note that the square root σ_h of $\mathrm{Var}[\mathcal{H}]$ is shown.

D. Trusted Candidate Distance Distribution

Since the metric of equation (10) is deterministic, it makes sense to see how far, in terms of edges, trusted candidates are located and how hyperparameter ρ_1 influences their distance distribution. The metric has been applied to all candidate vertices for each hyperparameter value and the mean was taken over all startups. Figure 2 shows only three of them (to avoid cluttering) for the cases $r = 1$, $r = 3$, and $r = 7$ in log scale. Index r is explained in table VIII. This not only separates better the curves but also reveals a pattern: Notice that for approximately the same values of the hyperparameter ρ_1 reported for table V the bulk of trusted candidates is concentrated in low distances. Outside this zone they tend to

TABLE VI
STATISTICS FOR THE TOTAL NUMBER OF STEPS DISTRIBUTION.

Statistics/ρ_1	0.1	0.2	**0.5**	1	2	5	10
I_h **(B/D)**	638.10/852.41	613.27/834.67	591.11/816.95	607.44/829.17	619.35/838.65	652.26/870.03	683.33/895.12
σ_h **(B/D)**	51.88/67.62	50.11/66.98	49.09/65.32	51.78/68.97	59.94/72.51	65.55/76.98	72.09/83.70
$\bar{\mu}_3[\mathcal{H}]$ **(B/D)**	0.30/0.15	0.31/0.18	0.33/0.21	0.31/0.16	0.28/0.07	0.25/−0.09	0.17/−0.15
$\bar{\mu}_4[\mathcal{H}]$ **(B/D)**	2.14/2.21	1.92/2.09	1.89/1.91	1.88/2.19	1.90/2.25	2.13/2.31	2.22/2.55

TABLE VII
GAUSSIANITY RESULTS FOR TOTAL NUMBER OF STEPS ($\rho_1 = 0.5$).

Test name	**BFS/DFS**	**Test name**	**BFS/DFS**
Kolmogorov-Smirnoff	No/No	Anderson-Darling	No/No
Lielliefors	No/No	Shapiro-Wilk	No/No

become more uniform. To assess this, table VIII has the values of the binary Kullback-Leibler divergence of (26) of distance distributions from the uniform distribution q:

$$\left\langle p^{(r)} \,||\, q \right\rangle \triangleq \sum_{l=1}^{L_0} p_l^{(r)} \log_2 p_l^{(r)} + \log_2 L_0 \tag{26}$$

Thus, when the relative trust weight is high there is a structure in the location of the candidates. Otherwise, their location disperses to completely random locations across the graph.

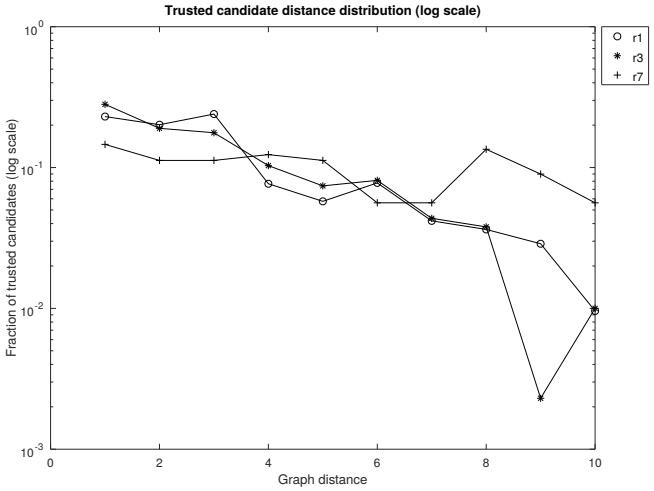

Fig. 2. Candidate distance distribution (log scale) vs distance.

E. Steps To First Trusted Candidate Distribution

Another evaluation metric for the proposed algorithms is the number of steps required to reach the first trusted candidate. The analysis of the r.v. $\mathcal{F}$ counting the mean over all startups is identical to that for the r.v. $\mathcal{H}$ of the total number of steps, so only the results and the conclusions will be given here.

As seen from table IX BFS requires fewer steps in general with a higher level of confidence. The skewness shows that the majority of the values are on the left side of the distribution, whereas kurtosis suggests that DFS has a heavier tail than BFS. Also note that again the square root σ_f of $\mathrm{Var}[\mathcal{F}]$ is shown so that it should be on the same scale with I_f.

F. Comments And Recommendations

Following [2] the findings can be explained by small world phenomena in the Patras startup ecosystem. This can be attributed to the many networking opportunities and to the systematic LinkedIn use as denoted by the high v_c to v_s ratio. Since trusted candidates are usually in the professional vicinity of the startup owners, local events should be attended.

The fact that for small relative trust weights candidate choices appear to be random implies that trust operates as a filter. Thus, candidates should seek opportunities to take credit for their work or engage in tasks allowing verifiable credit. Another way is frequent peer interaction, as this will lead to more endorsements and thus to indirect skill recognition.

Observe that all metrics take their best value when ρ_1 equals 0.5 or in that zone. This is clear indication that in the search for trusted candidates the balance between trust and skillset compatibility should favor the former but not by much.

V. CONCLUSIONS AND FUTURE WORK

This conference paper focuses on discovering trusted candidates for startups in LinkedIn graph with a BFS and a DFS strategy. Both have a vertex selection mechanism relying on skillset similarity and a match metric based on trust computed from LinkedIn attributes. The proposed algorithms were verified by on the startup growing ecosystem of Patras, Hellas. Higher order probabilistic analysis suggests the superiority of the BFS variant. The latter has been explained and recommendations were given based on these results.

This work can be extended in a number of ways. First, a conceptual tree for skills can be developed so that distance metrics like Leacock-Chodorow can be used. A higher order trust system based on attributes such as endorsements from coworkers or highly skilled members can improve our local approximation. Probabilistic analysis can include the distribution of steps between the discovery of two trusted candidates. The analysis as conducted here stands and can be carried over to other types of graph search algorithms as well.

ACKNOWLEDGMENT

This conference paper is partially funded by Research Initiative Fund Grant (RIF) 18056, Zayed University, UAE.

REFERENCES

[1] F. Muñoz-Bullon, M. J. Sanchez-Bueno, and A. Vos-Saz, "Startup team contributions and new firm creation: The role of founding team experience," *Entrepreneurship and Regional Development*, vol. 27, no. 1-2, pp. 80–105, 2015.

[2] T. Fonseca, P. de Faria, and F. Lima, "Human capital and innovation: The importance of the optimal organizational task structure," *Research policy*, vol. 48, no. 3, pp. 616–627, 2019.

TABLE VIII
DIVERGENCE OF TRUSTED CANDIDATE DISTANCE DISTRIBUTIONS.

Divergence/ρ_1	0.1	0.2	**0.5**	1	2	5	10
r	1	2	3	4	5	4	7
$\langle p^{(r)} \mid\mid q \rangle$	0.4602	0.4720	0.5374	0.4833	0.4584	0.4191	0.3540

TABLE IX
STATISTICS FOR THE STEPS TO FIRST TRUSTED CANDIDATE DISTRIBUTION.

Statistics/ρ_1	0.1	0.2	**0.5**	1	2	5	10
I_f **(B/D)**	17.33/20.45	14.88/17.69	9.11/12.38	13.01/16.08	16.73/19.27	17.87/22.21	24.55/29.05
σ_f **(B/D)**	12.33/13.25	10.37/10.47	7.21/9.93	9.86/10.72	10.30/11.01	12.74/13.66	13.38/14.56
$\bar{\mu}_3[\mathcal{F}]$ **(B/D)**	0.27/0.25	0.26/0.24	0.29/0.25	0.27/0.23	0.26/0.20	0.23/0.15	0.17/−0.04
$\bar{\mu}_4[\mathcal{F}]$ **(B/D)**	1.26/1.11	1.29/1.15	1.31/1.17	1.29/1.14	1.26/0.89	1.21/0.81	1.11/0.69

[3] T. Aynaud and J.-L. Guillaume, "Static community detection algorithms for evolving networks," in *WiOpt*. IEEE, 2010, pp. 513–519.

[4] M. A. Russell, *Mining the social Web: Data mining Facebook, Twitter, LinkedIn, Google+, GitHub, and more*. O'Reilly Media, Inc., 2013.

[5] A. Auradkar *et al.*, "Data infrastructure at LinkedIn," in *ICDE*. IEEE, 2012, pp. 1370–1381.

[6] R. Sumbaly, J. Kreps, and S. Shah, "The big data ecosystem at LinkedIn," in *SIGMOD*, 2013, pp. 1125–1134.

[7] J. Zide, B. Elman, and C. Shahani-Denning, "LinkedIn and recruitment: How profiles differ across occupations," *Employee relations*, vol. 36, no. 5, pp. 583–604, 2014.

[8] G. Pinho, J. Arantes, T. Marques, F. Branco, and M. Au-Yong-Oliveira, "The use of LinkedIn for ICT recruitment," in *World Conference on Information Systems and Technologies*. Springer, 2019, pp. 166–175.

[9] A. Archambault and J. Grudin, "A longitudinal study of Facebook, LinkedIn, and Twitter use," in *SIGCHI conference on human factors in computing systems*, 2012, pp. 2741–2750.

[10] M. M. Skeels and J. Grudin, "When social networks cross boundaries: A case study of workplace use of Facebook and LinkedIn," in *International conference on supporting group work*, 2009, pp. 95–104.

[11] D. Garcia, K. M. Cloninger, A. Granjard, K. Molander-Söderholm, C. Amato, and S. Sikström, "Self-descriptions on LinkedIn: Recruitment or friendship identity?" *PsyCh journal*, vol. 7, no. 3, pp. 152–153, 2018.

[12] J. Guillory and J. T. Hancock, "The effect of LinkedIn on deception in resumes," *Cyberpsychology, behavior, and social networking*, vol. 15, no. 3, pp. 135–140, 2012.

[13] Y. He, R. Wang, X. Wang, J. Zhou, and Y. Yan, "Novel adaptive filtering algorithms based on higher-order statistics and geometric algebra," *IEEE Access*, vol. 8, pp. 73 767–73 779, 2020.

[14] U. Libal and K. H. Johansson, "Yule-Walker equations using higher order statistics for nonlinear autoregressive model," in *SPSympo*. IEEE, 2019, pp. 227–231.

[15] D. Rönnow and P. Händel, "Nonlinear distortion noise and linear attenuation in MIMO systems - Theory and application to multiband transmitters," *IEEE Transactions on Signal Processing*, vol. 67, no. 20, pp. 5203–5212, 2019.

[16] U. R. Acharya *et al.*, "Automatic detection of ischemic stroke using higher order spectra features in brain MRI images," *Cognitive systems research*, vol. 58, pp. 134–142, 2019.

[17] S. A. Khoshnevis and R. Sankar, "Applications of higher order statistics in electroencephalography signal processing: A comprehensive survey," *IEEE Reviews in biomedical engineering*, vol. 13, pp. 169–183, 2019.

[18] J. Craske, "Adjoint sensitivity analysis of chaotic systems using cumulant truncation," *Chaos, Solitons & Fractals*, vol. 119, pp. 243–254, 2019.

[19] G. Drakopoulos and P. Mylonas, "Evaluating graph resilience with tensor stack networks: A keras implementation," *NCAA*, vol. 32, no. 9, pp. 4161–4176, 2020.

[20] D. J. Cook and L. B. Holder, *Mining graph data*. John Wiley & Sons, 2006.

[21] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of massive data sets*. Cambridge University Press, 2020.

[22] T. Schank and D. Wagner, "Finding, counting and listing all triangles in large graphs, an experimental study," in *International workshop on experimental and efficient algorithms*. Springer, 2005, pp. 606–609.

[23] T. Horváth, T. Gärtner, and S. Wrobel, "Cyclic pattern kernels for predictive graph mining," in *ICDM*, 2004, pp. 158–167.

[24] Z. Kang, L. Wen, W. Chen, and Z. Xu, "Low-rank kernel learning for graph-based clustering," *Knowledge-Based Systems*, vol. 163, pp. 510–517, 2019.

[25] H.-T. Wai, S. Segarra, A. E. Ozdaglar, A. Scaglione, and A. Jadbabaie, "Blind community detection from low-rank excitations of a graph filter," *IEEE Transactions on signal processing*, vol. 68, pp. 436–451, 2019.

[26] G. Drakopoulos, P. Gourgaris, and A. Kanavos, "Graph communities in Neo4j: Four algorithms at work," *EVOS*, June 2018.

[27] R. Ibrahim and D. Gleich, "Nonlinear diffusion for community detection and semi-supervised learning," in *The WWW Conference*, 2019, pp. 739–750.

[28] R. Chen, J. Shi, Y. Chen, B. Zang, H. Guan, and H. Chen, "Powerlyra: Differentiated graph computation and partitioning on skewed graphs," *ACM Transactions on parallel computing*, vol. 5, no. 3, pp. 1–39, 2019.

[29] G. Drakopoulos, F. Stathopoulou, A. Kanavos, M. Paraskevas, G. Tzimas, P. Mylonas, and L. Iliadis, "A genetic algorithm for spatiosocial tensor clustering: Exploiting TensorFlow potential," *EVOS*, January 2019.

[30] J. B. Lee, R. A. Rossi, S. Kim, N. K. Ahmed, and E. Koh, "Attention models in graphs: A survey," *ACM Transactions on knowledge discovery from data*, vol. 13, no. 6, pp. 1–25, 2019.

[31] G. Drakopoulos, A. Kanavos, P. Mylonas, and S. Sioutas, "Defining and evaluating Twitter influence metrics: A higher-order approach in Neo4j," *SNAM*, vol. 7, no. 1, pp. 52:1–52:14, 2017.

[32] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, "A survey of deep learning-based network anomaly detection," *Cluster computing*, pp. 1–13, 2019.

[33] S. Rose, D. Engel, N. Cramer, and W. Cowley, "Automatic keyword extraction from individual documents," *Text mining: Applications and theory*, vol. 1, pp. 1–20, 2010.

[34] A. A. Salatino *et al.*, "The computer science ontology: A large-scale taxonomy of research areas," in *International Semantic Web Conference*. Springer, 2018, pp. 187–205.

[35] A. Kipf *et al.*, "Estimating cardinalities with deep sketches," in *ICDMa*, 2019, pp. 1937–1940.

[36] G. Drakopoulos, S. Kontopoulos, and C. Makris, "Eventually consistent cardinality estimation with applications in biodata mining," in *SAC*. ACM, 2016.

[37] A. Tversky, "Features of similarity," *Psychological review*, vol. 84, no. 4, p. 327, 1977.

[38] B. Metcalfe, "Metcalfe's law after 40 years of Ethernet," *Computer*, vol. 46, no. 12, pp. 26–31, 2013.

[39] T. T. Cai, T. Liang, and H. H. Zhou, "Law of log determinant of sample covariance matrix and optimal estimation of differential entropy for high-dimensional Gaussian distributions," *Journal of Multivariate Analysis*, vol. 137, pp. 161–172, 2015.

[40] P. Druilhet, D. Pommeret *et al.*, "Invariant conjugate analysis for exponential families," *Bayesian Analysis*, vol. 7, no. 4, pp. 903–916, 2012.

[41] N. M. Razali, Y. B. Wah *et al.*, "Power comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors, and Anderson-Darling tests," *Journal of statistical modeling and analytics*, vol. 2, no. 1, pp. 21–33, 2011.