Noname manuscript No.
(will be inserted by the editor)

Provenance Network Analytics

An approach to data analytics using data provenance

Trung Dong Huynh · Mark Ebden · Joel Fischer · Stephen Roberts · Luc Moreau

Received: date / Accepted: date

Abstract Provenance network analytics is a novel data analytics approach that helps infer properties of data, such as quality or importance, from their provenance. Instead of analysing application data, which are typically domain-dependent, it analyses the data's provenance as represented using the World Wide Web Consortium's domain-agnostic PROV data model. Specifically, the approach proposes a number of network metrics for provenance data and applies established machine learning techniques over such metrics to build predictive models for some key properties of data. Applying this method to the provenance of real-world data from three different applications, we show that it can successfully identify the owners of provenance documents, assess the quality of crowdsourced data, and identify instructions from chat messages in an alternate-reality game with high levels of accuracy. By so doing, we demonstrate the different ways the proposed provenance network metrics can be used in analysing data, providing the foundation for provenance-based data analytics.

Keywords data provenance · data analytics · network metrics · graph classification

We gratefully acknowledge funding from the UK Engineering and Physical Sciences Research Council (EPSRC) for project ORCHID, grant EP/I011587/1, and for project SOCIAM, grant EP/J017728/2.
Data statement: The data used for the production of this article, along with the associated experiment code, is publicly available at `https://github.com/trungdong/datasets-provanalytics-dmkd`.

T.D. Huynh
Electronics and Computer Science, University of Southampton
E-mail: tdh@ecs.soton.ac.uk

M. Ebden · S. Roberts
Information Engineering, Department of Engineering Science, University of Oxford

J. Fischer
Mixed Reality Lab, School of Computer Science, University of Nottingham

L. Moreau
Department of Informatics, King's College London

1 Introduction

Provenance, a description of what influenced the generation of a piece of information or data, has become an important topic in several communities since it exposes how information flows in systems, providing the means to make them accountable and helping users decide whether information is to be trusted (Moreau 2010). Provenance has been recorded in an increasing number of applications, from legal notices,[1] climate science (Ma et al 2014), medical applications,[2] scientific workflows (Alper et al 2013; Silva et al 2011; Davidson et al 2007; Altintas et al 2006), computational reproducibility (Chirigati et al 2013), emergency response (Ramchurn et al 2016), and in the geospatial domain.[3]

As a provenance description 'links' artefacts with their influences, it can be represented in a graph, called a *provenance graph*, whose nodes represent the artefacts/influences and whose edges their relations with one another. Studying such graphs, e.g. by visualising them, can facilitate understanding of the provenance information they contain. However, in a typical application, provenance graphs can quickly become very large and complex; this makes it difficult to interpret their information manually. For instance, as an indication, the 2014 edition of the United States' National Climate Assessment report[4] was published with full provenance information linking its data and recommendations to 242 authors and over 500 distinct technical inputs (Tilmes et al 2013). The scale is a few magnitudes larger with automated applications. CollabMap (Ramchurn et al 2013), an online crowd-sourcing platform, recorded more than 5,000 provenance graphs over three months running, many of which contain 30–200 nodes, 50–700 edges. Scientific workflows (e.g. Wolstencroft et al 2013; Silva et al 2011; Gil et al 2011; Bowers et al 2008) being applied to peta-scale problems, are also generating vast amount of provenance information. Such large and complex graphs are overwhelming for manual interpretation or verification (of data correctness, for instance). Therefore, an automated and principled way to analyse provenance data of such scales and, more importantly, to understand what they convey with respect to the data they describe, is much needed.

Against this background, in this paper, we propose *provenance network analytics*, a novel data analytics approach that combines network analysis and established machine learning techniques (Russell and Norvig 2010, Ch. 18) over provenance information generated automatically from log and instrumentation of applications. It provides a generic way to analyse provenance information with the aim of revealing real-word characteristics of the data about which it describes. Our contributions to the state-of-the-art are as follows:

1. First, we adapt a number of existing network metrics (Newman 2010) to suit provenance graphs and define provenance-specific ones to summarise the topological structure of provenance graphs. The *provenance network metrics* can be

[1] `https://www.thegazette.co.uk/`

[2] `https://www.hl7.org/fhir/provenance.html`

[3] `http://www.opengeospatial.org/projects/initiatives/ows-10`

[4] The online version of the report, provided with its provenance, is available at `http://nca2014.globalchange.gov/`

computed in a generic manner from provenance records and are independent of domain-specific information. Therefore, they provide the basis for analysing and/or comparing provenance graphs quantitatively, even those from different applications.
2. Second, we make use of the provenance network metrics to construct predictive models on provenance information based on known ground truths to relate provenance information with properties of data, such as their quality or importance. Once successfully trained for an application, those predictive models operate without relying on domain-specific information. By so doing, we devise a novel analytics method that analyses data using their provenance, *not* the data themselves. Thanks to the generic nature of the proposed provenance network metrics, our approach can be used to study data, via the means of their provenance graphs, in applications where provenance information is recorded.
3. Finally, we report the successful application of the above method on the provenance of real-world data from three different applications: identifying owners of provenance documents, assessing the quality of crowd-generated data in CollabMap, and identifying instructions from chat messages in an alternate-reality game. The applications were selected in part because they allow us to verify the accuracy of the proposed analytics via known ground truths or via an alternative method. In these applications, our analytic method achieved high levels of accuracy classifying data based on the provenance of such data. By so doing, we also demonstrate *how* the provenance network analytics approach can be concretely applied in specific contexts as a generic tool for data analytics.

The remainder of this paper is organised as follows. Section 2 introduces the provenance network metrics that serve as the basis for the provenance network analytics method presented in Section 3. Section 4 describes the evaluation methodology. In Section 5, we report on how the method was used to correctly identify owners of provenance graphs. Section 6 specialises the approach for quality assessment and demonstrates how the quality of crowd-generated data is classified. Section 7 shows that the same approach can help identify instructions from chat messages in the Radiation Response Game (Fischer et al 2014). We relate our approach to existing work in Section 8 and conclude the paper with directions for future work in Section 9.

2 Provenance Network Metrics

In this work, we adopt the PROV data model (Moreau and Missier 2013) as the data model for provenance in our analyses. PROV was standardised by the World Wide Web Consortium to support for the interchange of provenance information on the Web. It defines provenance as a "record that describes the people, institutions, entities, and activities involved in producing, influencing, or delivering a piece of data or a thing". The core PROV concepts are shown in Fig. 1. Owing to space limitations, the complete descriptions of those concepts could not be included here; the reader is encouraged to refer to Moreau and Missier (2013) for their full formal definitions. In brief, provenance records describe the generation and use of *entities* by some *activities*, which may be influenced in some ways by *agents*. Such records

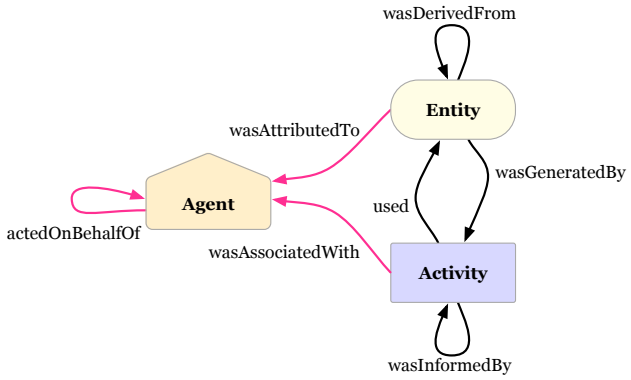

Fig. 1 The core PROV elements and relations (adapted from Lebo et al 2013).

can be packaged together into a *provenance document* for the purpose of exchanging provenance information.

Since provenance information describes how various elements were related to, or influenced by, one another, it can be viewed as a directed graph in which those elements (i.e. entities, activities, agents) are represented as nodes, and the relations between them (e.g. used, wasGeneratedBy, wasDerivedFrom) as directed edges. Such a graph is called a *provenance graph*. Given that some provenance graphs can be very large, the challenge is how to extract useful information and knowledge from complex provenance graphs. In that respect, we turn to the established field of graph theory for principled methods to analyse graphs. Specifically, we are interested in network metrics that allow us to summarise the topological characteristics of a provenance graph, such as its shape, its size, or how its nodes tend to connect to one another. Such network metrics are generic and can be calculated on any graphs, including provenance ones. They provide us a way to summarise provenance graphs into a set of generic network features. As a result, they allow for the comparisons of provenance graphs, even those from different domains or applications, without the need for the knowledge required to interpret domain-specific information contained therein.

In the following sub-sections, we enumerate the network metrics we employ for our analysis of provenance graphs and provide their formal definitions. Section 2.1 describes the generic network metrics which we adapted to work with provenance graphs. We then define provenance-specific network metrics in Section 2.2 to take advantage of provenance-specific information readily available in a provenance graph such as the types of nodes and the relations between them.

2.1 Generic Network Metrics

A provenance graph is a directed graph $G = (V_G, E_G)$, with vertex set V_G and edge set E_G. Vertices in V_G represent the PROV elements (i.e. entities, activities, and agents).

There is an edge $e = (v_i, v_j) \in E_G$ if there is a PROV relation in the graph relating vertex v_i to v_j, $v_i, v_j \in V_G$, in that direction. In addition, we define a function type that gives the provenance types of all vertices and edges in a provenance graph as follows:

$$\mathsf{ElementTypes} = \{\ \mathsf{Entity}, \mathsf{Activity}, \mathsf{Agent}\ \} \tag{1}$$

$$\begin{aligned}\mathsf{RelationTypes} = \{\ &\mathsf{Generation}, \mathsf{Usage}, \mathsf{Start}, \mathsf{End}, \mathsf{Derivation}, \mathsf{Invalidation},\\ &\mathsf{Communication}, \mathsf{Attribution}, \mathsf{Association}, \mathsf{Delegation},\\ &\mathsf{Membership}, \mathsf{Alternate}, \mathsf{Specialization}, \mathsf{Influence}\}\end{aligned} \tag{2}$$

$$\mathrm{type} = (V_G \rightarrow \mathsf{ElementTypes}) \cup (E_G \rightarrow \mathsf{RelationTypes}) \tag{3}$$

In order to summarise the topological characteristics of a provenance graph, we adopt common existing network metrics for this. Those metrics regard a provenance graph as an ordinary directed graph and, therefore, disregard provenance-specific information (which, however, will be later considered in Section 2.2). As a convention, the metrics in this section are defined on an input graph $G = (V_G, E_G)$ and for the sake of brevity we omit G where it is unambiguous. The generic network metrics included in our analyses are:

- Number of nodes $n = |V|$, which is also the number of provenance elements in G.
- Number of edges $e = |E|$, which is also the number of provenance relations in G.
- Graph diameter d_G is the longest *distance* in a graph G, where the distance between two vertices u and v is defined as the length of the shortest path between them, denoted by $d(u, v)$. The graph diameter reflects how "spread out" the provenance graph G is.

  $$d_G = \max_{u,v \in V_G} d(u, v) \tag{4}$$

  Since nodes in provenance graphs are separated by directed edges, thereby preventing some nodes from forming a path to certain others, strictly speaking, the diameter of each graph is, in many cases, infinite. However, by temporarily assuming the edges are undirected, we are able to calculate the diameter of a provenance graph. Hence, let $G^u = (V, E^u)$ be the undirected counterpart of G, i.e. whose edges are the same as those in G but undirected: $E^u = E \cup \{(v, u) \,|\, (u, v) \in E\}$. The diameter of a provenance graph G is then defined as d_{G^u}. For the sake of brevity, we simply use d to denote the graph diameter of G^u.
- Assortativity coefficient r: Assortativity, or assortative mixing, is the tendency for vertices in networks to be connected to other vertices that are like them in some way (Newman 2003). The assortativity coefficient is the Pearson correlation coefficient r of degree between pairs of linked nodes. Positive values of r indicate a correlation between nodes of similar degree, while negative values indicate relationships between nodes of different degree. r is defined as per Eq. 24 in Newman (2003).
- Average clustering coefficient ACC: The local clustering coefficient c_v of a vertex in a graph quantifies how close its neighbours are to being a clique (complete graph) (Watts and Strogatz 1998) and was introduced to determine whether a graph is a small-world network:

  $$c_v = \frac{\Gamma_v}{\deg_v (\deg_v - 1)} \tag{5}$$

where $\deg_v$ is the number of v's neighbours and Γ_v is the number of edges between the neighbours. The average c_v of all vertices in G represents the extent of neighbourhood clustering in G. In order to avoid biased assessment of the metric, following Kaiser (2008), we exclude leaf and isolated nodes (i.e. $\deg_v \leqslant 1$) in our calculation:

$$\mathsf{ACC} = \overline{C}, \text{ where } C = \left\{c_v | v \in V \wedge \deg_v > 1\right\} \tag{6}$$

– Degree distribution: For many real-world graphs, the degree distribution follows a 'power law' such that the number of vertices N_k with degree k is given by $N_k \propto k^{-\alpha}$, where $\alpha > 0$ is usually called the power-law exponent. We examine the degree distribution of an entire provenance graph to determine whether the distribution fits a power law as per the method of Clauset et al (2009) and, if so, the degree-distribution power-law exponent (DPE). For provenance graphs whose degree distribution does not fit a power law and α is therefore undefined, we manually set $\alpha = -1$.

2.2 Provenance-specific Network Metrics

In contrast to an ordinary directed graph, a provenance graph contains additional provenance type information on its the nodes and edges, as provided by the above type function (Eq. 3). In this section, we extend generic network metrics to characterise provenance graphs while taking provenance types into account, and by so doing, define a set of provenance-specific network metrics:

– Numbers of entities n_{e}, activities n_{a}, and agents n_{ag} in a provenance graph.
– Maximum finite distance (MFD): Since provenance relations represent a form of influence (Moreau and Missier 2013), the length of the longest chain of influence in a provenance graph is a useful characteristic of the graph. It can be viewed in the same vein as the graph diameter in the previous section but now on the directed graph G. In more detail, given two vertex sets $X, Y \subset V$, let $L_{X \to Y}$ the set of all finite distances separating a vertex in X with another in Y: $L_{X \to Y} = \{d(u, v) | u \in X \wedge v \in Y \wedge d(u, v) \neq \infty\}$. The MFD between X and Y in G, denoted as $\mathrm{mfd}_{X \to Y}$, is defined as follows:

$$\mathrm{mfd}_{X \to Y} = \begin{cases} -1 & \text{if } L_{X \to Y} = \emptyset \\ \max L_{X \to Y} & \text{otherwise} \end{cases} \tag{7}$$

As the kind of influence between different types of provenance elements is quite different, it is interesting to know the MFD between one node type and another. Considering only the distances between entities, for example, the MFD would reflect how far a piece of data was derived from, or somehow influenced by, another; while considering only the distances between agents might reveal how far delegation between them went. Since there are three different node types in a provenance graph, we define nine different MFD metrics, one for each pair of node types: $\mathrm{mfd}_{t_s \to t_e}$, $t_s, t_e \in \{\mathsf{e}, \mathsf{a}, \mathsf{ag}\}$, where e, a, and ag are our shorthand notation to denote V's subsets whose elements are Entity, Activity, and Agent, respectively.

Table 1 Glossary of provenance network metrics

Metric name	Symbol	Variants	Number of metrics
Number of elements	n and n_t	$t \in \{\mathsf{e}, \mathsf{a}, \mathsf{ag}\}$	4
Number of relations	e		1
Graph diameter	d		1
Assortativity coefficient	r		1
Average clustering coefficients	ACC and ACC_t	$t \in \{\mathsf{e}, \mathsf{a}, \mathsf{ag}\}$	4
Degree-distribution power-law exponent	α		1
Maximum finite distance	$\mathrm{mfd}_{t_s \to t_e}$	$t_s, t_e \in \{\mathsf{e}, \mathsf{a}, \mathsf{ag}\}$	9
Maximum finite distance of derivations	$\mathrm{mfd}_{\mathsf{der}}$		1

– MFD of derivations ($\mathrm{mfd}_{\mathsf{der}}$): Since Derivation is the only influence relation in PROV that has a strict time ordering (Cheney et al 2013), we calculate additionally the MFD over this relation to examine the longest chain of derivations in a provenance graph. For this, we consider $G_{\mathsf{der}} = (V, E_{\mathsf{der}})$, where $E_{\mathsf{der}} = \{e | e \in E \wedge \mathrm{type}(e) = \mathsf{Derivation}\}$, i.e. the sub-graph that contains only Derivation relations from G.

$$\mathrm{mfd}_{\mathsf{der}} = \begin{cases} -1 & \text{if } E_{\mathsf{der}} = \emptyset \\ \max\left\{ d_{G_{\mathsf{der}}}(u, v) | u, v \in V \wedge d_{G_{\mathsf{der}}}(u, v) \neq \infty \right\} & \text{otherwise} \end{cases} \tag{8}$$

where $\mathrm{mfd}_{\mathsf{der}}$ is set to -1 if there is no derivation relation in the graph.

– Average clustering coefficients by node type (ACC_t): This is a variation of the ACC metric (6); it is calculated from the local clustering coefficients of vertices of a given node type $t \in \{\mathsf{e}, \mathsf{a}, \mathsf{ag}\}$:

$$\mathsf{ACC}_t = \overline{C}_t, \text{ where } C_t = \left\{ c_v | v \in V \wedge \mathrm{type}\,(v) = t \wedge \deg_v > 1 \right\} \tag{9}$$

As there are three different provenance node types, there are also three provenance-specific ACC metrics: ACC_e, ACC_a, and ACC_ag.

2.3 Summary

Combining the generic and provenance-specific network metrics (see Table 1 for a summary), for a given provenance graph G, its provenance network metrics $\mathcal{P}(G)$ are represented in a vector containing twenty-two elements:

$$\begin{aligned} \mathcal{P}(G) = \langle\ & n, & n_\mathsf{e}, & n_\mathsf{a}, & n_\mathsf{ag}, & e, & d, \\ & r, & \mathsf{ACC}, & \mathsf{ACC}_\mathsf{e}, & \mathsf{ACC}_\mathsf{a}, & \mathsf{ACC}_\mathsf{ag}, & \alpha, \\ & \mathrm{mfd}_{\mathsf{e}\to\mathsf{e}}, & \mathrm{mfd}_{\mathsf{e}\to\mathsf{a}}, & \mathrm{mfd}_{\mathsf{e}\to\mathsf{ag}}, & \mathrm{mfd}_{\mathsf{a}\to\mathsf{e}}, & \mathrm{mfd}_{\mathsf{a}\to\mathsf{a}}, & \mathrm{mfd}_{\mathsf{a}\to\mathsf{ag}}, \\ & \mathrm{mfd}_{\mathsf{ag}\to\mathsf{e}}, & \mathrm{mfd}_{\mathsf{ag}\to\mathsf{a}}, & \mathrm{mfd}_{\mathsf{ag}\to\mathsf{ag}}, & \mathrm{mfd}_{\mathsf{der}} & \rangle \end{aligned} \tag{10}$$

In the next section, the above provenance network metrics serve as the basis for analysing provenance graphs to infer properties of the data they describe.

3 Provenance Network Analytics

The provenance network metrics defined in Section 2 can help us summarise a provenance graph's topological characteristics and allow for the quantitative comparison of provenance graphs. The metrics can tell us a graph with $n = 4$, $e = 3$, and $d = 3$ (i.e. a linear graph), for example, has a much different shape compared to a graph with $n = 4$, $e = 3$, and $d = 2$ (i.e. a star graph). However, without the ability to relate those values to domain-specific interpretation, say, the former is the result of a valid run while the latter is not, the network metrics alone would not help us to gain useful information contained in such graphs.

In this respect, we propose to apply existing supervised learning methods (see e.g. Russell and Norvig 2010, Ch. 18) to provenance graphs, using their network metrics as the features to predict some domain-specific characteristics of the data or events described by the graphs. The method requires a set of labelled training data, i.e. provenance graphs for which their classifications are known. The network metrics of those are then used as examples to train a predictive model for the interested classification. In essence, such a model predicts the label of a *whole* provenance graph from its network metrics. If it can be shown that the model has a high predictive power given the training data, it can later be used for classifying unseen provenance graphs from the same domain. In more detail, the approach consists of three main phases:

- **Design**: The purpose of this phase is to define the classification problem and to curate the required training data.
  1. Define the classification labels: This step formalises the classification problem into a discrete set of labels $\mathcal{L}$. Given a piece of data x from the application domain, the classification problem becomes that of predicting the label of x: $l_x \in \mathcal{L}$. For example, if we want to determine whether an application run is valid or not, we could have $\mathcal{L} = \{valid, invalid\}$; if we want to assess the quality of a data entity, we could have $\mathcal{L} = \{good, bad, uncertain\}$.
  2. Define the input provenance graph: Since we aim to use the provenance network metrics as inputs for a predictive model, we need to have the provenance graph of x to produce the metrics. As a provenance graph can record provenance of multiple entities, spanning from a few relations to the full history of an application run, choosing an appropriate extent of the input provenance graph G_x of x such that it sufficiently covers x's related history to be considered but not too broad, is a key decision. If the chosen provenance graph is too small, it may not include relevant relations that could determine the label of x; on the contrary, a too broad provenance graph, would have redundant information (i.e. noise) that could confuse a learning algorithm. Some knowledge of the application domain is useful for this step. In the above example of application-run validity, for instance, one might choose the whole provenance graph recorded from one run as the input, while a much smaller graph covering the generation and usages of a data entity might be more appropriate for the assessment of its quality. Concrete examples of this step are later provided in Sections 5, 6, and 7, with the last showing how this step can be automated.

3. Curate training data: As this method relies on supervised learning techniques, a curated set of labelled training data $S = \{(x, l_x) | l_x \in \mathcal{L}\}$ is required (i.e. l_x is defined for all x in S).

– **Training**: Having defined the label set $\mathcal{L}$, defined the input provenance graph G_x for all x to be classified, and curated the training data set S, we build the predictive model which is, in essence, a function that maps $\mathcal{P}(G_x)$ (Eq. 10) to $\mathcal{L}$:
  1. Choose a supervised learning algorithm[5] that suits $\mathcal{L}$ and the given data set.
  2. Calculate the network metrics for the provenance graphs of the labelled data and transform them into feature vectors with classification labels suitable as inputs to the chosen learning algorithm: $I = \{(\mathcal{P}(G_x), l_x) | (x, l_x) \in S\}$.
  3. Assess the accuracy of the learning algorithm on the input labelled data I.
  4. If the accuracy in obtained in Step 3 is sufficiently high,[6] build the classifier for $\mathcal{L}$ from I with the chosen learning algorithm and proceed to the Prediction phase.
– **Prediction**: Use the classifier from the Training phase to predict the labels of unseen data from their provenance.

4 Empirical Evaluation

As a tool for data analytics, the provenance network analytics method aims to discover correlations between provenance information and properties of the data it describes. In order to demonstrate the approach, we apply the method to the provenance of real-world data from three different applications and report its performance in the following sections. Before doing that, however, we first describe the common methodology for evaluating the method.

Learning algorithm: We use the CART (Breiman et al 1984) algorithm to train decision tree classifiers (specifically the Scikit-learn implementation by Pedregosa et al 2011). Empirically, we find decision tree classifiers perform sufficiently well and were fast, although not always producing the highest accuracy. For the three selected applications, other learning algorithms we tested could only marginally improve classification accuracy while incurring significant increases in computing cost, in many cases several magnitudes higher, compared to that of the decision tree classifier (see the Extra 1 experiment in the online Supplementary Materials for more details). In addition, a decision tree classifier is able to explain its classification with decision rules, which may provide useful clues to understand the correlation between an application's provenance data and the interested data properties.

Balancing learning data: To avoid producing biased classifiers, for datasets whose samples are unbalanced, we balance the input dataset I using the SMOTE method (Chawla et al 2011), which oversamples the minority samples such that each label has roughly the same number of samples in I.

[5] Since the field of supervised learning is broad and it is not the focus of this paper, the reader is suggested to refer to Russell and Norvig (2010) and Marsland (2014) for an overview of the available learning algorithms and their suitability for a specific dataset.

[6] The required level of accuracy depends on the intended application of the predictive model.

Assessing accuracy: In order to benefit from all the available labelled data, which are small in some cases, we use 10-fold cross-validation (Kohavi 1995). In particular, with I randomly split into 10 equal subsets, we perform 10 rounds of learning; on each round a $1/10$ subset is held out as the test set and the remaining are used as training data. To further minimise the potential chance impact of random data splitting, we repeat the above cross-validation procedure 100 times, hence collecting 1,000 accuracy scores in each experiment. We report the mean accuracy score alongside its 95% confidence interval in parentheses, for example, 98.13% (±0.01%).

In addition to the accuracy of classifiers, we also evaluate the following.

Relevance of metrics: From each round of learning in the cross-validation procedure, the trained classifier automatically calculates the relevance of each input feature (i.e. each of the 22 network metrics in Eq. 10) given the training data. In practice, this information will help us selectively reduce the number of metrics to be considered within a specific application (if required). We report the three most relevant network metrics for each application, i.e. those with the highest average relevance values.

Generic vs provenance-specific metrics: We repeat the above process (i.e. the Training phase and evaluation) in two further experiments — one using only the generic network metrics (Section 2.1) and the other only the provenance-specific network metrics (Section 2.2). Comparing the mean accuracy scores from the two experiments will help understand whether the network metrics based on provenance types bring added benefits to the classification application being discussed.

In the following sections, we report the exercise of provenance network analytics to build provenance-based classifiers for three different applications: identifying the owner of provenance documents on ProvStore (Huynh and Moreau 2015) (Section 5), assessing the quality of crowdsourced data in CollabMap (Section 6), and identifying instructions from chat messages in the Radiation Response Game (Section 7). In each application, since the Prediction phase, in which a classifier is run on unseen data, is straight-forward, we will not discuss it but will focus on the Design and Training phases of the method, which are to be followed by an evaluation as outlined above. The datasets and code to produce the results and figures in the following sections are provided with this article in the Supplementary Materials (see Appendix A for more details).

5 Application 1: Identifying Owner of Provenance Documents

As the PROV data model provides a vocabulary for provenance information, it is likely that each provenance producer has its own "style" of writing provenance using the vocabulary. For example, a user or an application may produce provenance graphs with chains of derivations that can be long or short, with or without attributions to agents, etc. Such individual styles will manifest in different topological characteristics of the resulting graphs and, hence, differences in the graphs' provenance network metrics. Our hypothesis is that the metrics could be used to identify the user or the application that produced a provenance graph. In order to verify this, we analyse the provenance network metrics of provenance documents deposited by the public at ProvStore, which is a public repository for provenance documents where a user can sign up for an account

and store their provenance online for sharing or visualisation purposes (Huynh and Moreau 2015). We apply the provenance network analytics method of Section 3 on those provenance documents to check how well it is able identify the documents' owners, here used as a proxy for the application that generated the provenance. Note that those provenance graphs typically do not contain any information about the users who uploaded the graphs to ProvStore, so it is not possible to identify such users simply from querying the graphs.

5.1 Design phase

Graph labels: We define the label set $\mathcal{L} = \{u_1, u_2, \ldots, u_n\}$, where $l_x = u_i$ if the provenance document x belongs to user u_i and n is the total number of users.
Input graphs: Since we want to identify the owner of a provenance graph based on its characteristics, we use the whole graph as the input graph, i.e. $X = x$.
Training data: In order to upload a provenance document to ProvStore, the document's owner needs to register for a user account there. As a result, the owner of each document on ProvStore is known and, hence, a curated labelled data set containing all those documents is readily available. Since each user owns a different number of documents, in order to ensure that there are sufficient samples to represent a user's provenance documents the Training phase, we limit our experiment to users who have at least 20 documents. There are fourteen such users (the authors were excluded to avoid bias), who we named $u_1, u_2, \ldots, u_{14}$; hence, there are 14 labels in $\mathcal{L}$. Their numbers of documents range between 21 and 6,745, with the total number of documents in the data set is 13,870.

5.2 Training phase

As described in Section 4, we train a single decision tree classifier to identify the owner of a given provenance document from the dataset. The 10-fold cross validation shows that the classifier can identify owners of provenance documents on ProvStore with a mean accuracy of 98.13% (±0.01%), compared to the baseline of 7.14% from selecting a random label from 14. This result strongly supports our hypothesis that the provenance network metrics represents the "signature" of provenance graphs, reflecting how the user or application that produces them models and records provenance information.

5.3 Discussion

The above result confirms the predictive power of provenance network metrics in analysing and identifying provenance graphs. The classifier itself, however, is of limited utility since we already knew the owners of all documents on ProvStore. Having said that, this is not necessary the case in applications where provenance data come from multiple, potentially unreliable, sources. In such cases, the provenance network analytics method could potentially help identify strange or suspicious

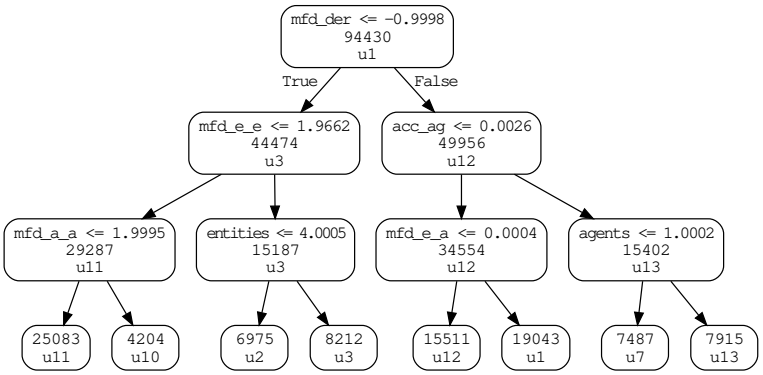

Fig. 2 The 3-depth decision tree for identifying owners of ProvStore documents.

provenance traces for further investigation, akin to graph-based anomaly detection techniques (Akoglu et al 2015). For instance, as provenance traces reflect the behaviour of an actor, the method can detect behaviours that are significantly different from the typical, which might represent an intrusion in cyber security contexts.

After the Training phase, a decision tree classifier is able to explain its classification rules in the form of a decision tree. As an example, the decision tree for identifying document owners above is shown in Fig. 2, whose depth, however, was limited to three to fit the paper. From the decision tree, it is apparent that the most influential metrics selected by the algorithm are provenance-specific ones. The most important metrics, in this case, is mfd_{der}; the tree splits the documents on ProvStore into two subsets: ones without derivation relation (i.e. $\text{mfd}_{\text{der}} = -1$, see Eq. 8) and ones with at least one derivation (the right branch). The tree shows the next most important metrics to distinguish provenance documents in this dataset are $\text{mfd}_{e \to e}$ and $\mathsf{ACC}_{\mathsf{ag}}$. From such information, we can see that half of the selected ProvStore users did not record derivations in their provenance at all. We can also learn that, for example, provenance documents uploaded by user u_2 contains no derivation, has 4 or fewer PROV entities, and the distances between those entities, if any, are less than 2. In addition, knowing which metrics are most relevant within an application or a dataset, one can make informed decision on which features that can safely be ignored (to save computation cost) in an automated manner, e.g. as per the method by Kohavi and John (1997).

In order to confirm that provenance type information indeed contributes into the above classification performance as suggested by the decision tree in Fig. 2, we repeat the exercise, training a decision tree with the same dataset, but this time with only the six generic network metrics (i.e. n, e, d, r, ACC, α) and later only the provenance-specific metrics. Compared to the first experiment, which makes use of all the available network metrics, using only the generic metrics achieves a lower accuracy at 92.32% (±0.02%), while using only the provenance-specific metrics produces a similar accuracy at 98.11% (±0.01%). These results suggest that provenance type information, as captured by the provenance-specific network metrics, indeed helps with identifying the originator of a provenance graph, and ignoring such information will result in a lower performance. Nevertheless, even with only six generic network

metrics, the trained classifier still achieves a very high level of accuracy (92.32%). Therefore, we believe that characterising provenance information by their network metrics is a very promising approach, which can be effective even with a small set of metrics. In the next section, we develop this approach further to assess the usage of crowd-generated data to infer about their quality, using only their provenance information.

6 Application 2: Assessing the Usage of Crowdsourced Data

The provenance of a piece of data tells us the *history* that led to its creation. Analysing its provenance may help ascertain the data's origin and that its production process was appropriate. It is, however, more challenging to infer about the data's quality or significance from its history without knowing the quality, reliability, or trustworthiness of the data's originator(s). Thus, instead of examining the data's *historical* provenance, we propose an alternative approach that examines the data's "forward provenance" — the records of how the data is used following its creation (also captured in an application's provenance traces). In this section, we apply the provenance network analytics on such "forward provenance" of crowdsourced data from CollabMap (Ramchurn et al 2013) to analyse their usage and, ultimately, their quality.

CollabMap is a crowdsourcing platform for constructing evacuation maps for urban areas. These maps need to contain evacuation routes connecting building exits to the road network, while avoiding physical obstacles such as walls or fences, which existing maps do not provide. The application crowdsources the drawing of such evacuation routes from the public by providing them with two sources of information from Google Maps: aerial imagery and ground-level panoramic views. It allows inexperienced users to perform tasks without them needing expertise other than drawing lines on a photo and does not rely on having experts verify the tasks in order to generate meaningful results. The task of identifying routes for a building was broken into different micro-tasks performed by different contributors: building identification (outline a *building*), building verification (vote for the building's validity), route identification (draw an evacuation *route*), route verification (vote for validity of routes), and completion verification (vote on the completeness of the current *route set*). This allows individual contributors to rate and correct each other's contributions (i.e. buildings, routes, and route sets). They were, however, not allowed to verify their own work to avoid biases.

In order to support auditing the quality of its data, the provenance of crowd activities in CollabMap was fully recorded: the data entities that were shown to users in each micro-task, the new entities generated therein, and their inter-relationships (see Fig. 3 for a small example with two micro-tasks). In 2012, CollabMap was deployed to help map the area around the Fawley Oil refinery in the United Kingdom. It generated descriptions for 5,175 buildings, 4,997 routes, and 4,710 route sets. In this application, we apply the provenance network analytics method to construct three classifiers in order to assess the quality of CollabMap data from their provenance, one for each type of data.

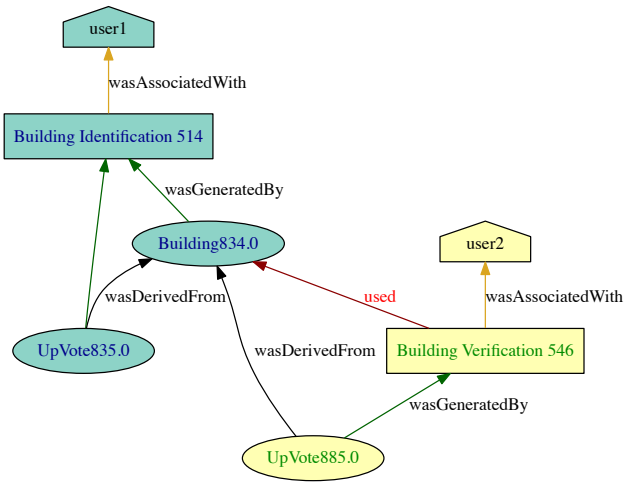

Fig. 3 A PROV graph from CollabMap.

6.1 Design phase

Graph labels: The main aim of assessing the quality of CollabMap data is to determine which of them are sufficiently trustworthy to be included in the final evacuation map. For a data entity x, we wanted to know whether x can be trusted to be correct ($l_x = \textit{trusted}$) or we are unsure about its quality ($l_x = \textit{uncertain}$). Thus, we define the label set $\mathcal{L} = \{\textit{trusted}, \textit{uncertain}\}$.

Input graphs: Whereas Application 1 had provenance graphs deposited separately and, hence, discretely split, CollabMap ran continuously and generated data whose provenance is interwoven with one another's. As a result, a CollabMap provenance graph contains the provenance of many data entities. As mentioned above, we propose to analyse the usage of a piece of data to infer its significance; for a given data entity x, we extract the provenance graph that contains all the activities and entities that were influenced by x. The intuition of the approach is similar to that of evaluating a publication from its citations. A highly cited academic paper, for example, is generally considered of high value thanks to its citations, or in other words, the relations it has with other papers. Such relations show how many times the paper was used in the generation of, or had an influence on, later papers.

Since a relation in PROV, i.e. an edge in our provenance graphs, represents some form of influence between its source and its target (Moreau and Missier 2013), if there exists a path in a provenance graph G from node v_i to node v_0, denoted as $v_i \xrightarrow{*} v_0$, then v_i was, in some way, potentially influenced by v_0. In other words, we consider here the transitive (potential) influence of v_0. It is possible to extract a sub-graph $D_{G,x} = (V_{G,x}, E_{G,x})$ from graph G containing only the nodes that were directly or

indirectly influenced by a particular node x, as follows:

$$V_{G,x} = \left\{ v \in V | v \xrightarrow{*} x \right\} \cup \{x\} \tag{11}$$

$$E_{G,x} = \left\{ e \in E | \exists v_s, v_t \in V_{G,x} \left(e = (v_s, v_t) \right) \right\} \tag{12}$$

We call $D_{G,x}$ the *dependency graph* of x extracted from the provenance graph G, or the transitive closure of x's potential influence in G; $V_{G,x}$ and $E_{G,x}$ are its vertex set and edge set, respectively. Hence, it is now possible to analyse the influence of x in G by examining the dependency graph $D_{G,x}$, which records how x was used in the application. Our hypothesis is that studying the dependency graph of x will reveal properties of x such as its value or quality. Hence, in CollabMap, we use the dependency graph of x as the input provenance graph in our quality analysis: $X = D_{G,x}$.

Training data: For the Training phase, we need to have a curated set of labelled training data. With the large amount of data generated in CollabMap, it was impractical to have them checked by experts. Collabmap instead relied on its participants to verify each other's work: buildings, evacuation routes, and route sets were cross-checked by the participants multiple times. The validity of buildings, routes, and route sets was ascertained by giving those entities either positive or negative votes. From the votes recorded, following the TRAVOS trust model (Teacy et al 2006), we define the trustworthiness of an entity x based on the beta family of probability density functions as follows:

$$\tau(x) = \frac{\alpha}{\alpha + \beta} \tag{13}$$

where $\tau(x)$ is the trust value of x (the mean of the beta distribution defined by the hyper-parameters α and β) with $\alpha = p + 1$ and $\beta = n + 1$; p and n are the numbers of positive and negative votes of x, respectively. Using the trust value $0 < \tau(x) < 1$, the label l_x for any data entity x in CollabMap can now be assigned as follows:

$$l_x = \begin{cases} \textit{trusted} & \text{if } \tau(a) \geqslant 0.75 \\ \textit{uncertain} & \text{otherwise} \end{cases} \tag{14}$$

where 0.75 is the threshold we chose to select data that were highly trusted by CollabMap's participants. Before proceeding to the Training phase, we calculated the trust values of all buildings, routes, and route sets and assign them with corresponding labels as specified by Eq. 14.

6.2 Training phase

We followed the methodology in Section 4, using decision tree classifiers and the same training and evaluation process. In this application, we trained three classifiers to classify the quality of buildings, routes, and route sets, one for each CollabMap data type. The accuracy of the classifiers is presented in Table 2 with the 95% confidence intervals and the number of samples available for each data type. The results show that the classifiers trained on the provenance network metrics of dependency graphs predict the trust labels for buildings, routes and route sets in the test sets with a high level of accuracy: 90% for buildings, 97% for routes, and 96% and route sets.

Table 2 CollabMap data quality classification results.

Data Type	Trusted	Uncertain	Accuracy	95% Confidence Interval
Building	4,491	684	90.03%	±0.06%
Route	3,908	1,089	96.98%	±0.04%
Route Set	3,019	1,691	95.70%	±0.05%

6.3 Discussion

With such high accuracy levels achieved by the classifiers, it is important to note that our method did not rely on any domain-specific information from CollabMap but only on generic, domain-independent provenance network metrics. The strong correlation between the provenance network metrics and data quality in CollabMap discovered by the classifiers suggests that analysing network metrics of provenance graphs is a promising approach to making sense of the (real-world) activities and data they describe, such as classifying crowd-generated data into trust categories as in this case. The use of provenance network analytics in applications like CollabMap could potentially reduce significantly the number of required verification tasks (which incur a cost in resources and/or time). In such cases, only a much smaller set of verification tasks would need to be carried out to generate enough training data for building the quality classifiers as shown above. While the provenance of a piece of data is traditionally examined to study its history, the successful application of provenance network analytics over "forward provenance" to analyse data's usage and significance in CollabMap shows that this can be an alternative useful approach for provenance analytics.

Relevance of metrics: As with the previous application, we also calculated the relevance of the network metrics after the Training phase. Although they were all generated from the same application, the most relevant metrics for their classification are quite varied: r, ACC, n_a for buildings; ACC, d, $\mathrm{mfd}_{\mathsf{der}}$ for routes; and r, $\mathsf{ACC}_{\mathsf{e}}$, e for route sets. Such differences are understandable given that buildings, routes, and route sets were used differently to create new data in CollabMap. Although the decision trees and the most relevant metrics above do not explicitly account for the connections between the features (i.e. the network metrics) and the prediction categories (i.e. the trust labels), they provide us with some starting points to help identify such connections in a later investigation.

Generic vs provenance-specific metrics: We retrained the three classifiers first using only the generic network metrics and later using only the provenance-specific metrics. The results (provided in Fig. 4) show that the classifiers trained only on the generic network metrics performed better than those trained only on the provenance-specific metrics in classifying buildings and routes but not route sets. However, the highest accuracy in this application were achieved by making use of the full set of provenance network metrics across the three data types.

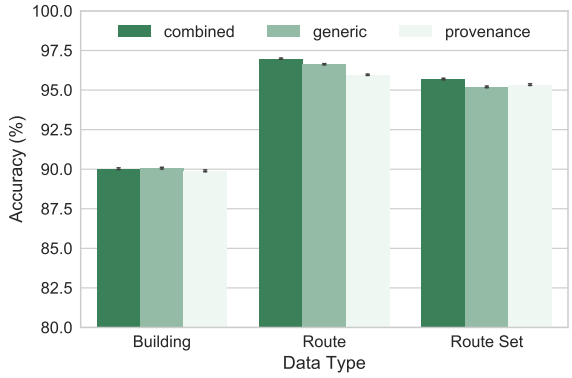

Fig. 4 The accuracy of quality classifiers for CollabMap buildings, routes, and route sets learned from generic and/or provenance-specific network metrics. Combining all the available metrics achieved the best classification accuracy.

7 Application 3: Identifying Instruction Messages

In the previous application, we introduce a method to extract the dependency graph of an entity of interest from a bigger provenance graphs to analyse its usage after creation. In this section, we show how the method can be further optimised to achieve the highest classification performance, in this case, inferring the significance of a chat message in the Radiation Response Game (RRG) (Fischer et al 2014) from its "forward provenance".

RRG is a location-based, mixed-reality game that simulates a disaster-response scenario in order to study team coordination. In this game, several spatially distributed targets (victims, animals, fuel, and resources) need to be recovered and moved to a safe place. Assisted by a headquarters, field responders (i.e. medics, fire-fighters, transporters, and soldiers) coordinate and form teams to move as many targets to safe places as possible. Each field responder communicates with the headquarters and the others via a smart phone app. The app also tracks the actions taken by responders such as picking up, moving, and dropping off a target, in addition to their locations. In order to assist the analysis of team coordination in a RRG, we record what happened in a game in a provenance graph which contains the following information:

- Agents: the field responders participating in the game and the headquarters.
- Entities: the targets and the messages communicated in the game.
- Activities: sending a message, picking up, transporting, and dropping off a target.

A small example of a RRG provenance graph is provided in Fig. 5, which shows one game activity in which `Animal121` was picked up by two field responders, generating a new provenance entity, `PickedUpAnimal121.1`, which is its new version with the updated status. A RRG provenance graph describes all the activities in a RRG, and, hence, is a large graph,[7] covering the evolution of the whole game and how its players and targets changed over time.

Since RRG was designed to study team coordination, the communications among participants are of particular interest as they can reveal when teams are formed and

[7] The RRG provenance graph used later in this section has 1,682 nodes and 4,184 edges.

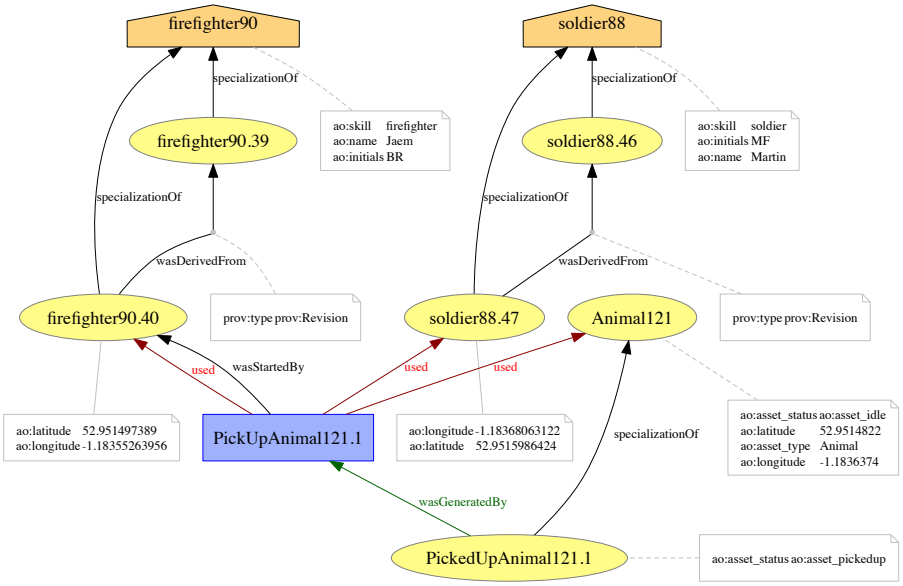

Fig. 5 Part of a PROV graph generated from a RRG capturing changes in one game activity.

what led to their formation. Therefore, in addition to automatically tracked game logs, each participant's voice communication is also recorded by individual recorders and their actions captured by video cameras. In a typical RRG game, there are eight to ten audio streams (one per responder), and four video cameras capturing the actions of the headquarters and the field responders over 30 minutes. Hence, post-hoc analysis of these audio and video recordings to learn about when and how team coordination happened requires significant human efforts. In practice, Fischer et al (2014) relied on the chat messages as a source to identify when teaming decisions were made and where to focus their investigation in the audio and/or video recordings. In order to do so, they first manually classified the chat messages into a number of different categories, the most interesting being that of *directives* as their study aims to determine whether, when, and why an instruction is followed or rejected (Fischer et al 2014). Our intuition is that such instruction messages, either followed or rejected, would lead to various activities by the game's participants following the moment the messages were received. For example, the participants could do as they were instructed, or they could send back more messages either to reject the instruction or to request further information. Since a RRG provenance graph also captures those activities, we believe that analysing the "forward provenance" of a chat message can help identify its role in the game. Hence, in what follows, we seek to apply the provenance network analytics method to identify instruction messages from their dependency (provenance) graphs.

7.1 Design phase

Graph labels: For each message in a RRG, Fischer et al (2014) classified them into one of the six categories: directives, assertives, expressives, declarations, commissives, and requests. Directives in RRG are typically instructions from the headquarters allocating tasks to the responder teams on the ground and are the targets for the classifier. Therefore, we label a message with *directive*, if it is one, or *other*, otherwise: $\mathcal{L} = \{directive, other\}$.
Input graphs: The dependency graph of a chat message in a RRG graph captures the activities that followed the message and, intuitively, is a suitable candidate to analyse to categorise a message. However, since a RRG graph evolves linearly along the time-line of a RRG, the size of such a dependency graph varies greatly depending on when in a game the message was sent; messages sent at the beginning of a game have significantly more (potential) dependants than those sent later in the game. In order to assess the *immediate* "impact" of a message, we limit the dependency graph of a message x to at most k edges away from the message in a RRG provenance graph. We called such dependency graph $D^k_{G,x} = (V^k_{G,x}, E^k_{G,x})$ such that:

$$V^k_{G,x} = \left\{v \in V | v \xrightarrow{k} x\right\} \cup \{x\} \tag{15}$$

$$E^k_{G,x} = \left\{e \in E | \exists v_s, v_t \in V^k_{G,x} \, (e = (v_s, v_t))\right\} \tag{16}$$

where $v \xrightarrow{k} x$ is true if there exists a path in G from v to x whose length is at most k. For a given k and a message x, we define the input graph $X = D^k_{G,x}$.
Training data: We recorded a single provenance graph for the RRG game reported by Fischer et al (2014), where there were 69 messages sent, 32 of which were categorised as directives. The dataset for this application is, hence, relatively balanced between the two labels (46% v.s. 54%); hence, no data re-balancing was carried out. For the training, each directive message is labelled as *directive*, while the rest as *other*.

7.2 Training phase

We carried out the training phase for this application in a similar manner as in the two previous applications. The main difference is that dependency graphs of messages are parameterised by k, the depth of the dependency graphs to be analysed. In order to determine the optimal value of k for this application, we have the training carried out with different values of k from 1 to 18.[8] The cross validation procedure then informs us the value of k that yields the best classification performance for our intended application. At the same time, this provides us with an insight into how far the "impact" of a message could be in the whole RRG provenance graph. Thus, for each $k \in [1, 18]$, we extracted $D^k_{G,x}$ for each message, and calculated the provenance network metrics for it. We then proceed with the training of a classifier to predict the label of a message x from its dependency graph. The mean accuracy of the classifier at each value of k and the confidence interval are plotted in Fig. 6.

[8] The accuracy level declines with $k > 15$ and, thus, we stop the exploration at $k = 18$.

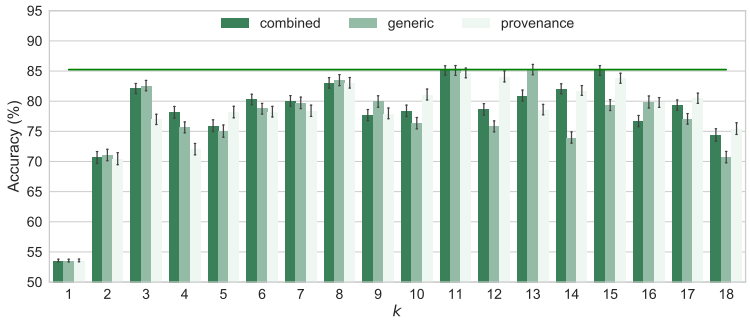

Fig. 6 The accuracy of the instruction message classifiers (of various dependency depth k) trained only on generic metrics, provenance-specific metrics, and both. The highest accuracy levels (marked by the horizontal line), 85%, were achieved with: $k = 11$ (combined/generic/provenance), $k = 13$ (generic), and $k = 15$ (combined). The accuracy level decreases with $k > 15$.

The results show that the classifier correctly identified directive messages on an average three out of four times for $k \geq 3$. It performed slightly worse at $k = 2$ and did no better than the base line at $k = 1$, suggesting that little or no useful network information was contained in such shallow dependency graphs. The exploration procedure discovers that $k = 11$ yields the top performance at 85.13% (±0.79%).

7.3 Discussion

In this application, we show how dependency graphs can be parameterised by their maximum depth (k) to help extract relevant input graphs for network analytics (in applications whose provenance graphs are too large and encompassing). The optimal value of k can be discovered in an automated manner by trialling different values and checking the classifier's performance with each value as shown earlier. Although the message classifier's accuracy is not as high as in the previous two applications, this level of accuracy (i.e. 85%) is sufficiently high for it to be useful as an automated analytic tool for the RRG study. It could assist post-hoc analysis of future RRG studies by labelling instructions from chat messages based on their provenance, significantly reducing manual efforts in classifying them as it has hitherto been done. Since our analytics method does not rely on the domain data but only on the provenance of activities in an RRG game, it can certainly be applied in other studies where similar provenance is captured. In those cases, the method can help identify points of interest in a study for further investigations, saving time and efforts of researchers going through voice and video recordings. In scenarios where instructions are already identified (e.g., tasks allocation, military orders), the result from this application suggests that analysing the "forward provenance" of such instructions could help determine their compliance with a predictive model.

Relevance of metrics: As with the previous applications, we examine the relevance of the network metrics; however, given the numerous configurations of k, we only give a summary of the results here. The detailed results are available in the Supplementary

Material. Across various values of k, we found that the most influential metrics was the number of edges e, followed by the number of entities n_e and $\text{mfd}_{e \to a}$. This is compatible with our earlier intuition that a directive message generally would generate more game activities, manifesting in more entities and provenance relations (i.e. edges) in the message's dependency graphs.

Generic vs provenance-specific metrics: Comparing the accuracy of classifiers trained only on the generic network metrics and that of those trained only on the provenance-specific metrics across the 18 values of k, the result is mixed. As shown in Fig. 6, both perform similarly in 7 cases, using provenance-specific metrics outperforms using only generic metrics in 7 cases, and in the remaining 4 cases, the reverse is true. It is difficult to draw a clear-cut conclusion from this. However, the result indicates that the provenance-specific metrics still plays a significant role in this application. Finally, both types of network metrics perform equally well in with $k = 11$, delivering the top accuracy for this application.

8 Related Work

Our work is conducted within the context of the descriptive analysis of network graph characteristics (Kolaczyk 2009). It has been shown that when studying a complex system such as a long-term crowdsourcing application or any program giving rise to a large amount of data (provenance or otherwise), various questions of interest can be rephrased usefully as questions regarding some aspect of the structure or characteristics of the corresponding network graph (Brandes and Erlebach 2005). For example, particular notions of the importance of individual system elements may be captured by measurements related to the corresponding vertices in the network. Indeed, Vaz de Melo et al (2012) provide a compelling example of why the inputs to a predictive algorithm should sometimes be based on network topology (such as those related to changes in the relationships among sports players and coaches) rather than node attributes (namely a player's performance statistics).

The field is continually evolving, and graphs can be viewed in a growing number of ways; provenance data itself can be interpreted as collaboration networks (Altintas et al 2010) or otherwise. Recently, Margo and Smogor (2010) examined a provenance graph based on components of a file-store, to show that provenance and other metadata can successfully predict semantic attributes: in particular, they predicted file extensions in a file-history graph. (As in earlier sections, 'predict' refers not to the temporal sense of the word, but to the re-inferring of removed data.) Although their particular choice of attribute to predict "has few applications", the study functioned as a useful proof of concept. The authors employed the C4.5 decision tree algorithm on their provenance graph, with the network structure and artefact attributes as input; the levels of accuracy achieved were comparable to our own, even though in the present work we examine provenance graphs of a different topology and size. The authors recognised that further exploration of the feature space over provenance graphs was called for; among other things, our methodology extends the types of features used in such analyses.

Related to our work in categorising provenance graphs, Cheah and Plale (2012) proposed a method to check for "structural flaws" in provenance graphs from workflow execution in order to detect anomalies. A component of the method relies on counting the number of nodes and edges from a set of provenance graphs to identify graphs that have too few or too many nodes/edges compared to their quantiles computed from the whole population. This approach, however, is only effective in the cases where provenance graphs are recorded from a workflow that is consistent in its outputs. Our approach, instead, employs machine learning techniques to reveal subtler and more complex correlations between such metrics and data properties. Moreover, it makes use of many more network metrics and also takes into account provenance type information. In order to cope with the potential complexity of provenance graphs (containing both structural information and node/edge provenance attributes), instead of directly analysing their network topology (like we did in this work), Chen et al (2014) proposed partitioning provenance graphs into subsets of vertices according to their temporal ordering. Scalar features (e.g. vertex type, the number of nodes in the subset, the average number of characters in node names) can then be collected for *each* subset of a provenance graph to represent the graph in tasks such as graph clustering, graph classifying, and rule mining. The key difference here is that the number of features can vary greatly depending how many subsets a provenance graph is partitioned into according to the Logical P algorithm by the same authors (Chen et al 2014). Therefore, the aforementioned data mining tasks can only be performed with graphs having the same number of subsets (i.e. the same number of features). Our provenance network metrics, on the contrary, are calculated on whole provenance graphs, and, hence, provide the same number of features regardless of graph size. In addition, the Logical P algorithm was designed to work with the Open Provenance Model (Moreau et al 2011) while our method was based on the later PROV Data Model (Moreau and Missier 2013) standardised by the World Wide Web Consortium. Given that not all PROV relations have temporal constraints associated with them, the Logical P algorithm may not work with certain PROV graphs.

Similar to our data quality assessment in Application 2 (Section 6), Ceolin et al (2014) sought to assess trustworthiness of crowdsourced data using provenance information. They relied, however, on node attributes (such as timestamps and typing speeds) rather than the network topology, in a different application area to ours (annotation of museum collections), achieving accuracies of approximately 80%. CrowdTruth (Inel et al 2014) is another crowdsourcing annotation application that sought to derive the quality crowdsourced data from a set of metrics on disagreements within the collected data. This work derived the metrics from the actual content of the data, not from analysing the relationships between them as per our method.

Our method provides a broader type of analysis than certain previous work on hyperlink network analysis (Park 2003) in which the links between web pages were studied to estimate the value of websites (e.g. their credibility) or to identify social networks. In the former case, the previous work only counted the number of links and did not investigate the network connections further than one link away (in contrast with the size of dependency graphs in our analyses). In the latter, the focus was on clustering similar nodes or detecting outliers, e.g. isolated nodes or those with few links, not on predicting node attributes as in this work. Also relatedly, Varlamis and

Louta (2009) count network links in order to express trustworthiness, as an example application of the principle described by Yu and Singh (2000) that propagation can be considered one of the properties of trust (along with symmetry, transitivity, self-reinforcement, etc.) However, Varlamis and Louta (2009) did not have voting data available in order to assess the accuracy of their model in the way we could as in the CollabMap application.

More generally speaking, graphs, as a generic and flexible data representation, are ubiquitous in describing computation. Analysing graph data is, hence, an active research topic of multiple communities in a variety of fields such as graph-based semi-supervised learning (Subramanya and Talukdar 2014), graph mining (Aggarwal and Wang 2010), and more. The latter includes frequent pattern mining (Cheng et al 2014), graph clustering (Gaertler 2005), graph classification (Tsuda and Saigo 2010), etc. Our work largely falls into the last area by providing a method for predicting the label of a *whole* provenance graph, as opposed to predicting the label of a node in the graph, also known as "label propagation" (Bengio et al 2006). However, compared to other graph classification techniques, our method makes use of network metrics instead of graph kernels (Vishwanathan et al 2010) or boosting (Saigo et al 2009); and the metrics were specifically constructed to work with PROV provenance graphs. Such provenance network metrics have not been studied before and our work is the first to propose employing them for characterising real-world properties of data in an automated manner.

9 Conclusions

Characterising properties of data, such as their quality or importance, can be challenging, especially with those generated by human contributors (like crowdsourced data or chat messages). It is usually a manual process that requires retrospection by experts who understand well the concerned application domain; in some other cases, it instead relies on the opinions of the participants (e.g. via a voting-like mechanism). In this work, we propose applying machine learning techniques on the network metrics of provenance graphs to explore and automate data characterisation. In particular, we have presented a generic and principled data analytics method for analysing data and applications based on their provenance graphs. Using this method, via the means of off-the-shelf machine learning algorithms, it is now possible to explore and learn about some properties of the data from their provenance in an automated manner. Since the method employs common network analyses and machine learning techniques on generic provenance graphs, it can be used in a wide range of applications where provenance are captured (or can be generated from application data/logs). Indeed, we have demonstrated the applicability of this method within three different applications: (1) identifying the owners of provenance documents on ProvStore, (2) classifying the trust labels for buildings, routes, and route sets drawn by crowd contributors in CollabMap, and (3) identify instructions from chat messages in the RRG;all resulted with high levels of accuracy. At the same time, we show how the method can be customised and optimised to suit a particular application context. Particularly, the results of Application 3 (Section 7) also led us to believe that the provenance

network analytics method can be a useful analytic tool for studying human activities or determining their compliance.

The twenty-two provenance network metrics we propose as features for analysing provenance information (Section 2) were chosen as the starting points of our investigation in this work. Although all of them were shown to contribute to the classification performance in the selected applications, each classification problem may still work well with a small subset of the metrics. Therefore, the relevance analysis is essential to identify those and to reduce the computation cost for unnecessary metrics. We plan to refine and develop further metrics from those twenty-two starting metrics. In particular, we are interested in refining the provenance-specific metrics to take into account the provenance semantics of Alternate and Specialization relations, which convey a different kind of influence than the others.

While avoiding using domain-specific information allows the provenance network metrics to be generically applied, we also appreciate the potential value of application-specific data in improving classification performance. In addition, certain applications may produce provenance graphs having the same topological characteristics, resulting the same set of network metrics values, confusing predictive models based solely on those. Therefore, in another future direction, we plan to extend our proposed metrics to utilise domain-specific information recorded in provenance information. We expect such customised provenance network metrics, albeit no longer generic, will help improve accuracy in analysing an application's data and will work with provenance graphs of highly similar topology.

In the three applications reported in this paper, we were able to collect the full provenance information recorded by them. This may, however, not always be the case. There can be applications where the provenance records available for analytics are incomplete or corrupted, or parts of them might be intentionally hidden or transformed to protect sensitive information (Cheney and Perera 2015; Danger et al 2015; Missier et al 2015; Hussein et al 2016). It is an open question how resilient a predictive model based on provenance network metrics performs against such variances in the input provenance graphs. An extension of this work, thus, could be on studying models of imperfect provenance information and their effects on provenance network analytics.

With an increasing number of applications continuously generating provenance,[9] we can quickly get overwhelmed with torrents of provenance data requiring our attention. The provenance network analytics method presented here can potentially be applied on provenance graph summaries, such as those produced by graph summarisation techniques (e.g. Moreau 2015; Riondato et al 2016), as significantly smaller proxies of the original provenance graphs, in order to lower computation costs. The analyses can also be extended to study the provenance network metrics that characterise the evolution of provenance graphs (like those introduced by Ebden et al 2012), which reflect the development of the tasks they represent. Such an extension could help us to understand developing dynamic behaviours, and to allow for appropriate on-the-fly interventions (in order to stop an undesirable behaviour from progressing, for instance).

[9] See `http://provenanceweek.org/2016/p3yl/programme.html` for examples.

In a wider context, provenance graphs do not only describe the origin of data, but they also reveal the interactions of agents in connected activities and how the activities themselves unfolded at the same time. The provenance network metrics presented in this work, therefore, could find useful applications in other areas in addition to those presented here. Analysing the influence of agents in the provenance graph of a collaborative task could identify the most valuable team member. Studying the distances between the agents in the graph could reveal close collaboration or team breakdown; or finding frequent patterns (Kuramochi and Karypis 2005; Yan et al 2008) in provenance graphs may show how they usually work together. In addition, focusing on the activities in a graph could help detect bottlenecks, important data, and activities that were crucial to the outcome of a task. Given the generic nature of network analysis techniques, the possibilities are highly promising and vast.

A Supplementary Materials

In order to help reproduce the results shown in this paper, we publish the datasets and the code used in the experiments reported above in the (electronic) Supplementary Materials, whose `README.md` file provides the full descriptions of the datasets and the code. In addition, the Supplementary Materials are also available online at `https://github.com/trungdong/datasets-provanalytics-dmkd`, where future updates and errata to the materials will be made.

References

Aggarwal CC, Wang H (2010) Graph data management and mining: A survey of algorithms and applications. In: Aggarwal CC, Wang H (eds) Managing and Mining Graph Data, Advances in Database Systems, vol 40, Springer US, Boston, MA, chap 2, pp 13–68, DOI 10.1007/978-1-4419-6045-0_2

Akoglu L, Tong H, Koutra D (2015) Graph based anomaly detection and description: A survey. Data Mining and Knowledge Discovery 29(3):626–688, DOI 10.1007/s10618-014-0365-y

Alper P, Belhajjame K, Goble CA, Karagoz P (2013) Enhancing and abstracting scientific workflow provenance for data publishing. In: Proceedings of the Joint EDBT/ICDT 2013 Workshops, ACM, New York, NY, USA, EDBT'13, pp 313–318, DOI 10.1145/2457317.2457370

Altintas I, Barney O, Jaeger-Frank E (2006) Provenance collection support in the Kepler scientific workflow system. In: Proceedings of the 2006 International Conference on Provenance and Annotation of Data, Springer, IPAW'06, pp 118–132, DOI 10.1007/11890850_14

Altintas I, Anand M, Crawl D, Bowers S, Belloum A, Missier P, et al (2010) Understanding collaborative studies through interoperable workflow provenance. In: Third International Provenance and Annotation Workshop, pp 42–58

Bengio Y, Delalleau O, Roux NL (2006) Label propagation and quadratic criterion. In: Olivier C, Schölkopf B, Zien A (eds) Semi-Supervised Learning, MIT Press, pp 193–216, DOI 10.7551/mitpress/9780262033589.003.0011

Bowers S, McPhillips T, Riddle S, Anand MK, Ludäscher B (2008) Kepler/pPOD: Scientific workflow and provenance support for assembling the tree of life. In: Freire J, Koop D, Moreau L (eds) Provenance and Annotation of Data and Processes, Lecture Notes in Computer Science, vol 5272, Springer Berlin Heidelberg, Berlin, Heidelberg, chap 9, pp 70–77, DOI 10.1007/978-3-540-89965-5_9

Brandes U, Erlebach T (2005) Network Analysis: Methodological Foundations. Springer

Breiman L, Friedman J, Olshen R, Stone C (1984) Classification and Regression Trees. Wadsworth, Belmont, California

Ceolin D, Nottamkandath A, Fokkink W (2014) Efficient semi-automated assessment of annotations trustworthiness. Journal of Trust Management 1(3):1–31, DOI 10.1186/2196-064X-1-3

Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2011) SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research 16:321–357, DOI 10.1613/jair.953

Cheah YW, Plale B (2012) Provenance analysis: Towards quality provenance. In: 2012 IEEE 8th International Conference on E-Science, IEEE, pp 1–8, DOI 10.1109/eScience.2012.6404480

Chen P, Plale B, Aktas MS (2014) Temporal representation for mining scientific data provenance. Future Generation Computer Systems 36:363–378, DOI 10.1016/j.future.2013.09.032

Cheney J, Perera R (2015) An analytical survey of provenance sanitization. In: Ludäscher B, Plale B (eds) Provenance and Annotation of Data and Processes. IPAW 2014, Lecture Notes in Computer Science, vol 8628, Springer, Cham, pp 113–126, DOI 10.1007/978-3-319-16462-5_9

Cheney J, Missier P, Moreau L, Nies TD (2013) Constraints of the PROV data model. W3C Recommendation REC-prov-constraints-20130430, World Wide Web Consortium, URL `http://www.w3.org/TR/2013/REC-prov-constraints-20130430/`

Cheng H, Yan X, Han J (2014) Mining graph patterns. In: Aggarwal CC, Han J (eds) Frequent Pattern Mining, Springer International Publishing, chap 13, pp 307–338, DOI 10.1007/978-3-319-07821-2_13

Chirigati F, Shasha D, Freire J (2013) Reprozip: Using provenance to support computational reproducibility. In: Proceedings of the 5th USENIX Conference on Theory and Practice of Provenance, USENIX Association, Berkeley, CA, USA

Clauset A, Shalizi C, Newman M (2009) Power-law distributions in empirical data. SIAM Review 51:661–703

Danger R, Curcin V, Missier P, Bryans J (2015) Access control and view generation for provenance graphs. Future Generation Computer Systems 49:8–27, DOI 10.1016/j.future.2015.01.014

Davidson SB, Boulakia SC, Eyal A, Ludäscher B, McPhillips TM, Bowers S, Anand MK, Freire J (2007) Provenance in scientific workflow systems. Data Engineering Bulletin 30(4):44–50

Ebden M, Huynh TD, Moreau L, Ramchurn S, Roberts S (2012) Network analysis on provenance graphs from a crowdsourcing application. In: Groth P, Frew J (eds) Provenance and Annotation of Data and Processes, Lecture Notes in Computer Science, vol 7525, Springer Berlin Heidelberg, pp 168–182, DOI 10.1007/978-3-642-34222-6_13

Fischer JE, Jiang W, Kerne A, Greenhalgh C, Ramchurn SD, Reece S, Pantidi N, Rodden T (2014) Supporting team coordination on the ground: Requirements from a mixed reality game. In: Rossitto C, Ciolfi L, Martin D, Conein B (eds) COOP 2014 - Proceedings of the 11th International Conference on the Design of Cooperative Systems, Springer International Publishing, Nice, France, pp 49–67, DOI 10.1007/978-3-319-06498-7_4

Gaertler M (2005) Clustering. In: Brandes U, Erlebach T (eds) Network Analysis, Lecture Notes in Computer Science, vol 3418, Springer Berlin Heidelberg, chap 8, pp 178–215, DOI 10.1007/978-3-540-31955-9_8

Gil Y, Ratnakar V, Kim J, Gonzalez-Calero P, Groth P, Moody J, Deelman E (2011) Wings: Intelligent workflow-based design of computational experiments. IEEE Intelligent Systems 26(1):62–72, DOI 10.1109/MIS.2010.9

Hussein J, Sassone V, Moreau L (2016) A template-based graph transformation system for the prov data model. In: Seventh International Workshop on Graph Computation Models GCM 2016

Huynh TD, Moreau L (2015) ProvStore: A public provenance repository. In: Ludäscher B, Plale B (eds) 5th International Provenance and Annotation Workshop, IPAW 2014, Lecture Notes in Computer Science, vol 8628, Springer International Publishing, Cologne, Germany, pp 275–277, DOI 10.1007/978-3-319-16462-5_32

Inel O, Khamkham K, Cristea T, Dumitrache A (2014) CrowdTruth: Machine-human computation framework for harnessing disagreement in gathering annotated data. In: Mika P, Tudorache T, Bernstein A, Welty C, Knoblock C, Vrandečić D, Groth P, Noy N, Janowicz K, Goble C (eds) The Semantic Web - ISWC 2014, Lecture Notes in Computer Science, vol 8797, Springer International Publishing, pp 486–504, DOI 10.1007/978-3-319-11915-1

Kaiser M (2008) Mean clustering coefficients: the role of isolated nodes and leafs on clustering measures for small-world networks. New Journal of Physics 10(8):083,042, DOI 10.1088/1367-2630/10/8/083042

Kohavi R (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, vol 2, pp 1137–1143

Kohavi R, John GH (1997) Wrappers for feature subset selection. Artificial Intelligence 97(1-2):273–324, DOI 10.1016/S0004-3702(97)00043-X

Kolaczyk E (2009) Statistical Analysis of Network Data. Springer

Kuramochi M, Karypis G (2005) Finding frequent patterns in a large sparse graph. Data Mining and Knowledge Discovery 11(3):243–271, DOI 10.1007/s10618-005-0003-9

Lebo T, Sahoo S, McGuinness D (2013) PROV-O: The PROV ontology. Tech. Rep. REC-prov-o-20130430, World Wide Web Consortium, URL https://www.w3.org/TR/2013/REC-prov-o-20130430/, W3C Recommendation

Ma X, Fox P, Tilmes C, Jacobs K, Waple A (2014) Capturing provenance of global change information. Nature Climate Change 4(6):409–413, DOI 10.1038/nclimate2141

Margo D, Smogor R (2010) Using provenance to extract semantic file attributes. In: Proceedings of the 2nd conference on Theory and practice of provenance, Berkeley, USA, USENIX Association

Marsland S (2014) Machine Learning: An Algorithmic Perspective. Chapman and Hall/CRC

Missier P, Bryans J, Gamble C, Curcin V, Danger R (2015) ProvAbs: Model, policy, and tooling for abstracting prov graphs. In: Provenance and Annotation of Data and Processes. IPAW 2014, Lecture Notes in Computer Science, vol 8628, Springer, Cham, pp 3–15, DOI 10.1007/978-3-319-16462-5_1

Moreau L (2010) The foundations for provenance on the web. Foundations and Trends in Web Science 2(2–3):99–241, DOI 10.1561/1800000010

Moreau L (2015) Aggregation by provenance types: A technique for summarising provenance graphs. In: Graphs as Models 2015 (An ETAPS'15 workshop), Electronic Proceedings in Theoretical Computer Science, Electronic Proceedings in Theoretical Computer Science, London, UK, pp 129–144, DOI 10.4204/EPTCS.181.9

Moreau L, Missier P (2013) PROV-DM: The PROV data model. Tech. Rep. REC-prov-dm-20130430, World Wide Web Consortium, URL http://www.w3.org/TR/2013/REC-prov-dm-20130430/, W3C Recommendation

Moreau L, Clifford B, Freire J, Futrelle J, Gil Y, Groth P, Kwasnikowska N, Miles S, Missier P, Myers J, Plale B, Simmhan Y, Stephan E, Van den Bussche J (2011) The Open Provenance Model core specification (v1.1). Future Generation Computer Systems 27(6):743–756, DOI 10.1016/j.future.2010.07.005

Newman M (2010) Networks: An Introduction. Oxford University Press

Newman MEJ (2003) Mixing patterns in networks. Physical Review E 67(2):026,126, DOI 10.1103/PhysRevE.67.026126

Park H (2003) Hyperlink network analysis: A new method for the study of social structure on the web. Connections 25(1):49–61

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

Ramchurn SD, Huynh TD, Venanzi M, Shi B (2013) CollabMap: Crowdsourcing maps for emergency planning. In: 5th ACM Web Science Conference (WebSci '13)

Ramchurn SD, Huynh TD, Wu F, Ikuno Y, Flann J, Moreau L, Fischer JE, Jiang W, Rodden T, Simpson E, Reece S, Roberts S, Jennings NR (2016) A disaster response system based on human-agent collectives. Journal of Artificial Intelligence Research 57:661–708, DOI 10.1613/jair.5098, URL http://www.jair.org/papers/paper5098.html

Riondato M, García-Soriano D, Bonchi F (2016) Graph summarization with quality guarantees. Data Mining and Knowledge Discovery pp 1–36, DOI 10.1007/s10618-016-0468-8

Russell S, Norvig P (2010) Artificial Intelligence: A Modern Approach, 3rd edn. Pearson

Saigo H, Nowozin S, Kadowaki T, Kudo T, Tsuda K (2009) gBoost: A mathematical programming approach to graph classification and regression. Machine Learning 75(1):69–89, DOI 10.1007/s10994-008-5089-z

Silva CT, Anderson E, Santos E, Freire J (2011) Using VisTrails and provenance for teaching scientific visualization. Computer Graphics Forum 30(1):75–84, DOI 10.1111/j.1467-8659.2010.01830.x

Subramanya A, Talukdar PP (2014) Graph-Based Semi-Supervised Learning, Synthesis Lectures on Artificial Intelligence and Machine Learning, vol 8. Morgan & Claypool, DOI 10.2200/S00590ED1V01Y201408AIM029

Teacy WTL, Patel J, Jennings NR, Luck M (2006) TRAVOS: Trust and reputation in the context of inaccurate information sources. Autonomous Agents and MultiAgent Systems 12(2):183–198

Tilmes C, Fox P, Ma X, McGuinness DL, Privette AP, Smith A, Waple A, Zednik S, Zheng JG (2013) Provenance representation for the National Climate Assessment in the Global Change Information System. IEEE Transactions on Geoscience and Remote Sensing 51(11):5160–5168, DOI 10.1109/TGRS.2013.2262179

Tsuda K, Saigo H (2010) Graph classification. In: Aggarwal CC, Wang H (eds) Managing and Mining Graph Data, Advances in Database Systems, vol 40, Springer US, chap 11, pp 337–363, DOI 10.1007/978-1-4419-6045-0_11

Varlamis I, Louta M (2009) Towards a personalized blog site recommendation system: A collaborative rating approach. In: Fourth International Workshop on Semantic Media Adaptation and Personalization, IEEE, San Sebastian, Spain, pp 38–43, DOI 10.1109/SMAP.2009.17, URL http://ieeexplore.ieee.org/document/5381709/

Vaz de Melo POS, Almeida VAF, Loureiro AAF, Faloutsos C (2012) Forecasting in the NBA and other team sports. ACM Transactions on Knowledge Discovery from Data 6(3):1–27, DOI 10.1145/2362383.2362387

Vishwanathan SVN, Schraudolph NN, Kondor R, Borgwardt KM (2010) Graph kenrels. Journal of Machine Learning Research 11:1201–1242

Watts DJ, Strogatz SH (1998) Collective dynamics of 'small-world' networks. Nature 393(6684):440–2, DOI 10.1038/30918

Wolstencroft K, Haines R, Fellows D, Williams A, Withers D, Owen S, Soiland-Reyes S, Dunlop I, Nenadic A, Fisher P, Bhagat J, Belhajjame K, Bacall F, Hardisty A, Nieva de la Hidalga A, Balcazar Vargas MP, Sufi S, Goble C (2013) The Taverna workflow suite: designing and executing workflows of web services on the desktop, web or in the cloud. Nucleic Acids Research 41(W1):W557–W561, DOI 10.1093/nar/gkt328

Yan X, Cheng H, Han J, Yu PS (2008) Mining significant graph patterns by leap search. In: Proceedings of the 2008 ACM SIGMOD international conference on Management of data, ACM, pp 433–444, DOI 10.1145/1376616.1376662

Yu B, Singh MP (2000) A social mechanism of reputation management in electronic communities. In: Cooperative Information Agents, Springer-Verlag Berlin Heidelberg, pp 154–165