applied sciences

MDPI

Article

User Trust Inference in Online Social Networks: A Message Passing Perspective

Yu Liu * and Bai Wang *

Beijing Key Laboratory of Intelligence Telecommunication Software and Multimedia, Beijing University of Posts and Telecommunications, Beijing 100876, China

* Correspondence: imyuliu@outlook.com (Y.L.); wangbai@bupt.edu.cn (B.W.)

Abstract: Online social networks are vital environments for information sharing and user interactivity. To help users of online social services to build, expand, and maintain their friend networks or webs of trust, trust management systems have been deployed and trust inference (or more generally, friend recommendation) techniques have been studied in many online social networks. However, there are some challenging issues obstructing the real-world trust inference tasks. Using only explicit yet sparse trust relationships to predict user trust is inefficient in large online social networks. In the age of privacy-respecting Internet, certain types of user data may be unavailable, and thus existing models for trust inference may be less accurate or even defunct. Although some less interpretable models may achieve better performance in trust prediction, the interpretability of the models may prevent them from being adopted or improved for making relevant informed decisions. To tackle these problems, we propose a probabilistic graphical model for trust inference in online social networks in this paper. The proposed model is built upon the skeleton of explicit trust relationships (the web of trust) and embeds various types of available user data as comprehensively-designed trust-aware features. A message passing algorithm, loop belief propagation, is applied to the model inference, which greatly improves the interpretability of the proposed model. The performance of the proposed model is demonstrated by experiments on a real-world online social network dataset. Experimental results show the proposed model achieves acceptable accuracy with both fully and partially available data. Comparison experiments were conducted, and the results show the proposed model's promise for trust inference in some circumstances.

Keywords: trust inference; trust propagation; online social network; social network analysis; probabilistic graphical model; message passing; belief propagation; model interpretability

check for updates

Citation: Liu, Y.; Wang, B. User Trust Inference in Online Social Networks: A Message Passing Perspective. *Appl. Sci.* **2022**, *12*, 5186. https://doi.org/10.3390/app12105186

Academic Editors: Dionisis Margaris and Stefanos Ougiaroglou

Received: 27 April 2022
Accepted: 18 May 2022
Published: 20 May 2022

1. Introduction

Trust exists in many different forms in various disciplines. For instance, the trust on the World Wide Web can be trust in content, trust in services, and trust in people [1]. Although different disciplines take different definitions and forms of trust, they commonly aim to solve the problem of accurately evaluating trust between two entities, to help complex systems make informed decisions. For example, some peer-to-peer system may take advantage of trust to curb malicious attacks and maintain network robustness [2]. A complex model selection system could evaluate the trustworthiness of a cloud-based machine learning model-as-a-service (MaaS) for industrial Internet of Things and smart city services [3]. Some online social service could use trust to improve the quality of recommendations [4]. Some researchers leveraged a trust network to study the relational antecedents of members' influence in organizations [5].

Without loss of generality, we focus on user trust in online social networks (OSNs) in this paper, which is one of the most common types of trust. We followed [6] to define user trust in OSNs: a subjective expectation an OSN user has about another user's future behavior. Social trust is a basic social construct [7], and it enables people to collectively live and work in groups. In social networks, trust helps people find whom to trust, to build beneficial or even reciprocal social relationships, so that the quality of interindividual interactions could be improved and the risks of social activities reduced.

Many online social services embrace trust management systems to help their users to build and expand their webs of trust so that users can keep being engaged in their services [8]. For example, Twitter has deployed a user recommendation service called "Who to Follow" to use a user's "circle of trust" for recommending new connections to the user [9]. Being a key contributing factor in many complex systems, trust has been elaborately explored and researched, and it has been proven to be helpful in securing social commerce [10], recommending trustworthy users [11], providing accurate and personalized recommendations [4,12–14], filtering trustworthy authorities or users [15], finding opinion leaders or trolls [16], maximizing influence diffusion [17], and decision making [18–20]. However, data of trust information, such as trust relationships, do not always explicitly or abundantly exist for the above-mentioned tasks to use. Therefore, the study of trust inference is necessary and practical for social network analysis and relevant decision making tasks.

The process of inferring an unknown trust relationship in OSNs, which is often referred to as trust inference, involves exploitation of social construct elements. The sources of relevant trust information that assembles social constructs in OSNs include but are not limited to existing trust relationships and data created by the users involved in the relationships, such as their activities, including posting reviews and casting votes, and other content generated by them (user-generated contents, UGC).

Various algorithms and models have been proposed to infer trust in OSNs [21]. Some of them make use of topological data, i.e., the web of trust relationships or the trust network, to predict trust relationships. However, due to the ever-present issue that observable trust relationships in OSNs are often sparse, some vanilla algorithms for predicting trust are prevented from achieving more accurate predicted results in large real OSNs. With the aim of tackling the issue, other methods use both the web of trust and UGC data to achieve more accurate results in inferring trust.

Nevertheless, there are three major problems holding back existing trust inference models. Firstly, if additional UGC data are available, some of them and various types of interplay among them are discarded when the trust inference framework integrates them with the trust network, making the model prone to generating less accurate results. Secondly, lesser types of data from OSN users are permitted to use, given the fact that the privacy and data protection of online users are being much more respected nowadays. Online service users may opt out of using some of or all of their data for certain data analysis tasks conducted by online services—especially with the regulations and laws being implemented and enforced, such as General Data Protection Regulation (EU) (GDPR) [22], the California Consumer Privacy Act (CCPA) [23], and the Personal Information Protection Law of the People's Republic of China (PIPL) [24]. Thus, some crucial data may be unobtainable for trust inference. Last but not least, most trust inference models are poorly interpretable. The interpretability of a trust inference model should be improved alongside achieving better performance, not only for inferring trust itself but for making other relevant informed decisions.

Bearing the aforementioned problems in mind, we propose creating a probabilistic graphical model in which various types of UGC data are built and then integrated as features in such a way that not only are most of their characteristics preserved, but that the interaction among features can also be captured and embedded. The contributions of this paper are as follows.

- The proposed model takes advantage of the integration of the trust network and user-generated contents in the network; the latter is embedded into a probabilistic graphical model built upon the former. The model permits the directionality of trust relationships and preserves various facets and properties of trust. The way of both

building features from UGC data and embedding them into the probabilistic graph preserves as much information as the data may contain.
- To infer trust, the proposed model uses a message passing algorithm, loopy belief propagation, for the model's probabilistic inference. This inference algorithm can be viewed as a reproduction of the propagative and incomplete transitive characteristics of trust. By using the message passing algorithm, the resulting probability for each predicted user-to-user trust relationship can be well interpreted.
- As a binary classification task, the performance of the proposed method to infer trust is demonstrated with a dataset derived from a real online social network in comparison with some state-of-the-art binary classifiers. Experimental results show the proposed model achieves better accuracy and F_1 score with the whole data presented and maintained higher recall and acceptable precision with some of data absent. Thus, one can conclude that the proposed model shows its promising ability for trust inference in nowadays privacy-constrained online social analysis where available data are often limited. To address the data limitation, the problem that a model should have higher precision or higher recall is also discussed.

Although this paper only focuses on inferring user trust in online social networks, the proposed model may also be adopted to fulfill other inference tasks to better assist decision making in other complex systems. The least work required for other similar tasks is to properly define a concrete graphical model and a set of reasonably-built features, if additional data exist.

The rest of the paper is organized as follows. In Section 2, we briefly review the literature of related works on trust inference. In Section 3, we elaborate the prerequisites, define the problem, and then propose the model. In Section 4, we present experiments conducted with a dataset derived from a real-world online social network, and then analyze the performance of the proposed model. Finally, we conclude our work in Section 5.

2. Related Work

The problem of trust inference or trust prediction between two users in online social networks or two nodes in general networks bears some similarity with the link prediction problem, and thus it can be modeled as a link prediction task. However, using a link prediction method to predict trust links requires the method to work with directional links or even signed links, which complicates the link prediction method itself.

For example, Schall [25] leverages micro-structural patterns and the resulting node similarity to retrieve the probability of a missing directional link existing between two users in a social network. The method's drawback is that the micro-structural pattern is limited to a triad, and consequently it may fail in finding links between any two arbitrary nodes that do not have a common neighbor.

The work by Barbieri et al. [26] also employed link prediction techniques to infer friend or trust relationships. In this work, they proposed a generative model, one of the stochastic topic models, to generate social links for users with the consideration of user's interests in "topical" and "social" resources, e.g., whether a targeted user is an authority on a topic or he/she is recognized by an acquaintance in the real world.

Trust inference models based on the trust propagation theory are often called "walk-based" methods. In [27], Mao et al. developed a trust inference framework which obtains the trust rate between any pair of users by aggregating a set of strong trust paths generated with the knowledge of their weighted similarity about commonly interesting topics and their trust propagation ability in the social network. If the trust rate is above a user-defined threshold, the framework determines that there should be a trust link between the users.

The work by Oh et al. [28] included a unified model combining both explicit and implicit trust, and infers trust links by using different trust propagation strategies. The three primary trust propagation strategies include direct propagation, transposed trust, and global trust propagation. Other complex strategies, such as co-citations and trust coupling, are combinations of the primary propagation strategies. Other walk-based methods include ModelTrust [29], TidalTrust [30], AssessTrust [31], OpinionWalk [32], etc.

Trust prediction can also be achieved and improved by using collaborative filtering techniques, particularly the matrix factorization (MF)-based methods. It is also quite convenient for MF-based methods to integrate other types of data that carry trust information. hTrust [33] incorporates low-rank matrix factorization and homophily regularization to infer trust links. The homophily regularization controls how user rating similarity affects predicting user trust relationships. MATRI [34] extracts user trust tendency using matrix factorization from the user trust rating matrix and incorporates trust propagation with trust tendency into a collective matrix factorization framework to infer trust. Another MF-based trust inference model [35] takes advantage of not only matrix-factorized trust tendency and propagated trust, but also similarities of users' rating habits, and achieves good performance.

Neural networks are also employed for trust evaluation. NeuralWalk [36] employs a neural network named WalkNet to model single-hop trust propagation, and then iteratively assesses unknown multi-hop trust relationships. Although NeuralWalk can achieve good accuracy in trust prediction, it is inefficient due to the massive matrix operations involved in training and test set selection. Besides, the interpretability for such methods based on neural networks is an obvious drawback.

3. The Proposed Model

In this section, we propose our model for user trust inference in OSNs. Firstly, we state the relevant assumptions and prerequisites on which the proposed model is based; secondly, we describe our modeling approach in detail; and finally, we briefly discuss the implementation of the model.

3.1. Prerequisites

A variety of studies and literature [6,21,37,38] have found that trust has many unique features and characteristics, and they are embedded in user profiles and other UGC data in OSNs. Based on relevant findings of trust and OSNs, we give some key principles and assumptions on which our proposed model stands in this section, without formal proofs.

Assumption 1. *Through activities of users in online social networks, user-generated content, such as reviews, posts, votes, issued trust, and so on, represent and bear the user's attitude, credibility, and trustworthiness.*

A survey [37] suggests that numerous factors, including logical trust attributes (e.g., experience, frequency, stability, and rationality), emotional trust attributes (e.g., hope, fear, and regret) and relational trust attributes (e.g., similarity), contribute to construct individual trust through social capital and social activities. Meanwhile, in online social networks, the above factors are expressed through various types of user-generated contents and user activities. Therefore, trust can be harvested through a variety of UGC data that exists in OSNs, such as their posted reviews or articles, cast ratings and votes, and issued trust relationships.

Assumption 2. *For a trust relationship between a trustor and a trustee, not only the trustee but also the trustor contributes to the relationship formation.*

The formation of a trust relationship involves the trustor who evaluates and issues trust, and the trustee who presents and receives it, which makes trust asymmetric and subjective. The issuance of a trust relationship is determined by the trustor through their perception of the trustee's trustworthiness and how others perceive it. However, due to the difficulty of modeling trust's subjectivity, many studies only focus on making use of

UGC data that represent trustees' credibility [39–41], leaving trust an objective concept and measure.

Based on this point, we bear the mind in this paper that although same types of user data are being used for collecting trust information, they could serve different purposes for the trustor and the trustee involved in a trust relationship. For example, the same characteristics extracted from one of a user's reviews may serve as different features: the type of attitude features that suggest how the user trusts others, or the type of trustworthiness features which indicate how he is being trusted by others.

Assumption 3. *Users bearing similarities in their profiles or activities have a higher likelihood to establish trust relationships.*

That is suggested by the homophily effect, which is one of the most important theories that attempt to explain why people establish trust relations with each other [42]. For example, in the situation of product reviewing, people with similar tastes about items are more likely to trust each other.

Taking the previous two assumptions into consideration, we could further state that a group of similar users may build trust relationships with similar users from another group, if there exists a trust relationship between a pair of users from each group. This also implies trust's incomplete transitivity property.

Assumption 4. *In datasets from online social networks, no observable or explicit trust relationship between two users does not truly guarantee there will not be any trust between them.*

For example, we may infer a trust relationship issued by Alice to Bob, provided that (1) there are many other users who have certain similar characteristics as Alice has, and they also trust Bob, or (2) Alice trusts many other users who have traits in common with Bob. However, it is worth noting that there are quite a few reasons that the relationship of Alice and Bob is not present in the network. It can be explained from three perspectives:

- Due to some particular or unknown facts, Alice does not trust Bob; the trust relationship does not exist, and therefore, it will never be observed.
- It might be possible that Alice would trust Bob at some time later, but at the time we observe the social network or capture a snapshot of the network as a dataset, Alice does not know Bob yet or Alice does not claim to trust Bob yet, so the trust relationship from Alice to Bob does not exist.
- Alice does trust Bob and the trust relationship does exist in the real network, but it is missing from the dataset we observe. The cause could be the inability of capturing the whole network or capturing their relationship data being prohibited by the privacy preference settings of relevant users.

This uniqueness of trust in OSNs makes trust inference in OSNs quite different from general link prediction problems and general binary classification tasks. Since we only focus on inferring trust relationships in online social networks without any distrust information, as a binary classification task, the goal of our proposed model is to find as many trust relations as possible, and in the meantime, maintain considerable overall accuracy. Further discussion on this assumption is beyond the scope of this paper, and it will be left for future work.

Assumption 5. *Trust is propagative but not fully transitive, so it will be beneficial for us to learn under what criteria trust can be transferable from one user to another.*

For example, if Alice trusts Bob and Bob trusts Chris, Alice can derive some amount of trust on Chris based on how much she trusts Bob and how much Bob trusts Chris, but Alice may not trust Chris or even she does not know Chris. However, under some particular circumstance which is what we need to learn, Alice will trust Chris. Based on this assumption, we propose two primary trust propagation strategies below so that our model can learn from more complex trust relationship topology that commonly exists in real-world datasets:

- Direct trust propagation may exist from Alice to Chris when Alice trusts Bob and Bob trusts Chris.
- Transposed trust propagation may exist from Alice to Chris when Alice trusts Bob and Chris also trusts Bob.

The proposed trust propagation strategies are demonstrated in Figure 1. Some more complex propagation schemes can be derived from the above two primary ones. For example, if Alice trusts Bob and Chris, and Daniel also trusts Bob and Chris, there might be an increase in the trust between Alice and Daniel, which is a cascaded result of two transposed trust propagation instances. In the field of citation network analysis, this very same scheme is called *co-citation* and has been greatly studied. It is worth noting that the co-citation propagation conforms to Assumption 3, but, differently from it, we derived the "co-citation"-equivalent trust propagation from the topology perspective. The same above-mentioned trust propagation strategies of direct trust propagation and transposed trust propagation were also leveraged in [28], which helped the researchers build a better trust prediction model.

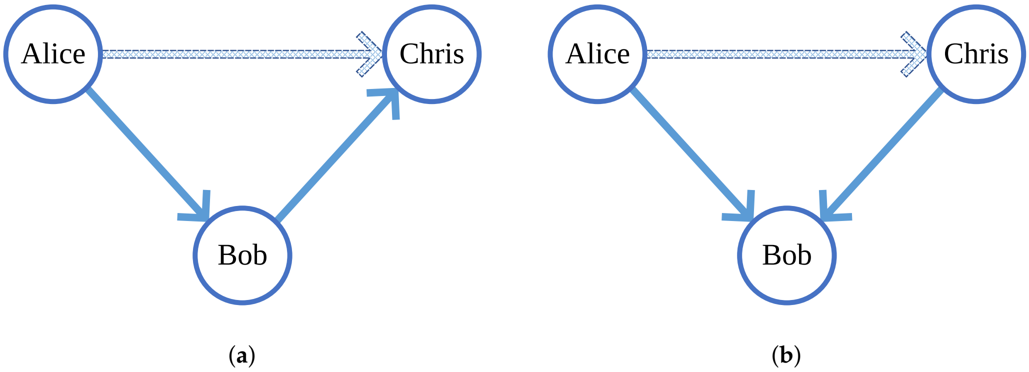


Figure 1. Two primary trust propagation strategies. (**a**) Direct trust propagation. (**b**) Transposed trust propagation.

3.2. Model Construction

Many modern online social services have deployed a feature which allows their users to build friend networks. It is common understanding that trust has a dedicated role in forming friendships between two individuals, and thus many trust-related studies also use friend networks in OSNs as trust networks. In particular, there are several online services that have explicitly implemented trust networks as web of trust—such online services include Epinion (http://www.epinions.com/ (accessed on 28 February 2018)) and Ciao (http://www.ciao.co.uk/ (accessed on 19 June 2021)).

We modeled our learning and inference tasks through a conditional random field (CRF), one of probabilistic graphical model variations. The structure of the probabilistic graphical model was built upon the trust network of an OSN, and the features that were to be added to the model were extracted and then built from the UGC data from the OSN. Due to the way that the UGC and topological features are embedded into the trust network, the CRF model not only uses both the explicit trust information (the trust network) and the user-generated contents from the social network, but also has the capability of capturing the interplay among various types of features extracted from the UGC and the network topology.

Differently from the conventional link prediction problem in which edges between nodes represent trust relationships, our method represents relationships as nodes. In other words, we model a real-world directional trust link as a trust relationship node in the model.

Therefore, the random variables in the CRF model are the predefined states or labels of users and trust relationships. (In this paper, we use “state” and “label” interchangeably.) In the CRF model, the two types of random variable nodes in the probabilistic graph are:

- **user** node. It can be either a *trustor* node or a *trustee* node;
- **trustRelation** node. It represents an observable or a nonexistent trust link in the network.

Figure 2 demonstrates the difference between a trust relationship commonly represented in real social networks and one modeled by our approach.

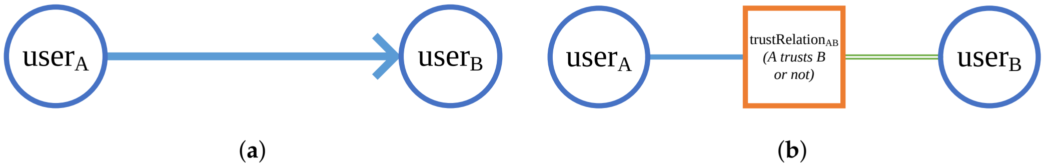


Figure 2. Demonstration of one user trusting another in the real world and in the proposed model. (**a**) Real-world trust relationship representation: user_A trusts user_B. (**b**) user_A trusting user_B rendered in the proposed model.

For a user-user pair whose trust relationship is un-observed in the OSN, the objective of the model in this paper is to infer the probabilities of labels of the corresponding *trustRelation* node whose state is unknown in the probabilistic graph, so that the trust relationship can be predicted.

It is worth looking at how we handle edges in the model. Edges in a CRF are usually homogeneous in type. That means edges in the graph do not have to be type-specific. However, without breaking any conventional rule for model inference, we may particularly mark edges with dedicated types so that different edges can delegate different types of meanings and thus serve different purposes. In our model, the type of an edge between a *trustor* node and a *trustRelation* node is different from the type of an edge between a *trustRelation* node and a *trustee* node. It is also worth noting that involving different edge types grant the model the possibility of recognizing directional trust between a pair of two users. An example of modeling the mutual relationship between two users is demonstrated in Figure 3.

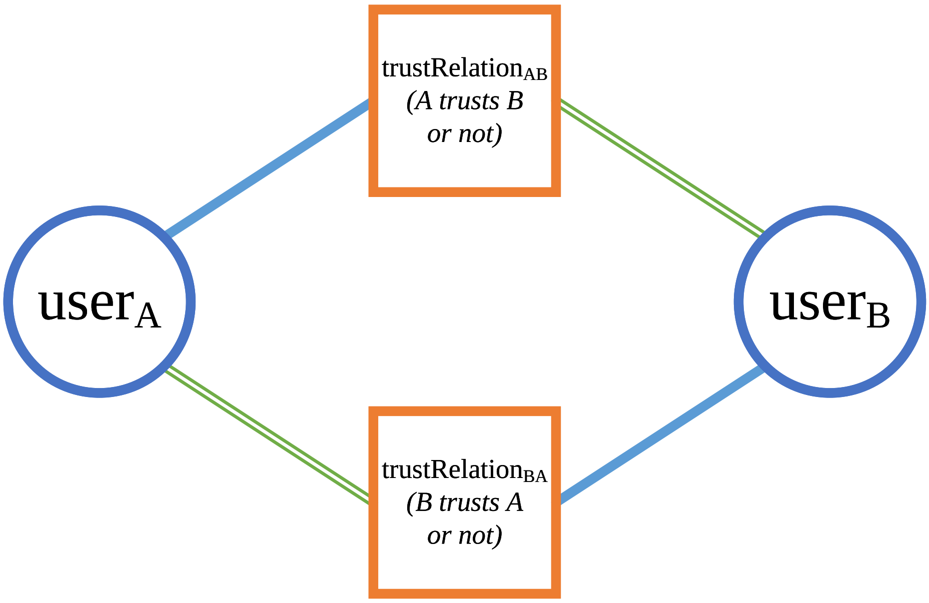


Figure 3. A demonstration of modeling two users’ mutual relationship in the proposed model.

3.2.1. Notation and Problem Definition

Let U_i denote the user random variable at node i and T_j or T_{ab} (or $\mathrm{T}_{a\to b}$) denote the trustRelation random variable at node j or the node representing the trust relationship from user U_a to user U_b. The notation of nodes and edges is detailed in Table 1.

Table 1. Notation of (random variable) nodes and edges in the proposed model.

	Notation	**Description**
Nodes		
	V	The set of all (random variable, r.v.) nodes.
user	U_u	R.v. for a user node u, either a trustor or a trustee.
trustRelation	T_t	R.v. for a trustRelation node t.
	T_{ab}	R.v. for a trustRelation node representing the trust relationship from user U_a to user U_b.
Edges		
	E	The set of all edges.
trustor ↔ trustRelation	E_{ut}	An edge between a trustor node U_u and a trustRelation node T_t.
trustRelation ↔ trustee	E_{tu}	An edge between a trustRelation node T_t and a trustee node U_u.

Now, we examine two users, for example, Alice and Bob, in a trust network and use a binary variable $\mathrm{T}_{\mathrm{Alice}\to\mathrm{Bob}}$ to represent the possible trust relationship from Alice to Bob. If a trust relationship from Alice to Bob is present in the network, it means that Alice trusts Bob and we label the variable with value 1. If no trust relationship from Alice to Bob is observed in the dataset, we label the variable with value 0. As trust is directional and asymmetry, a different variable $\mathrm{T}_{\mathrm{Bob}\to\mathrm{Alice}}$ also exists, and its value, which is also binary, represents the observational result of the trust relationship from Bob to Alice. Furthermore, we use probabilities to indicate trust relationships. For any two users A and B:

$$\begin{array}{ccc} P(\mathrm{T}_{\mathrm{A}\to\mathrm{B}}=1) > P(\mathrm{T}_{\mathrm{A}\to\mathrm{B}}=0) & , & \text{if A trusts B;} \\ \textit{otherwise} & , & \text{if A does not express trust in B.} \end{array} \tag{1}$$

Additionally, particularly for trust relationships being observed in the dataset:

$$\left.\begin{aligned} P(\mathrm{T}_{\mathrm{A}\to\mathrm{B}}=1) = \quad 1 \\ P(\mathrm{T}_{\mathrm{A}\to\mathrm{B}}=0) = \quad 0 \end{aligned}\right\} \quad , \quad \text{if A trusts B;} \tag{2}$$

$$\left.\begin{aligned} P(\mathrm{T}_{\mathrm{A}\to\mathrm{B}}=1) = \quad 0 \\ P(\mathrm{T}_{\mathrm{A}\to\mathrm{B}}=0) = \quad 1 \end{aligned}\right\} \quad , \quad \text{if A does not express trust in B.} \tag{3}$$

The notation can be easily extended to support distrust, a concept that differs from either trust or no trust being observed in the dataset. However, distrust is beyond the scope of this paper, hence we would like to leave it for future research.

With the notation, the trust inference problem can be stated as follows. Given all user nodes U and a set of observed existing and nonexistent trustRelation nodes (their probability representations are either $P(\mathrm{T}=1)=1$ or $P(\mathrm{T}=1)=0$), the method predicts a set of un-observed trust relationships T^* by comparing their probability representations $P(\mathrm{T}^*=1)$ and $P(\mathrm{T}^*=0)$ that are calculated during the model inference.

3.2.2. Features

As stated in Assumption 1, trust information can be obtained, and trust can be harvested and evaluated through a user's generated contents in online social networks, including reviews, posts, connections, etc., as they are the representation and bearer of the user's attitude, credibility and trustworthiness, i.e., the trust constructs. Therefore, we extract various types of features from the dataset and embed them to the probabilistic graphical model to aid trust inference.

In our conditional random field, an arbitrary number of features of any arbitrary type can be attached to any node or any edge. All the features used in the proposed model are discrete, and they can be *label-observation features* for nodes, *label-label-observation features* for edges or *label-label features* for edges. In other words, the feature function of each feature is non-zero for a single state per node or a single state per edge (the state of an edge is determined by the state-pair of the two nodes connected by the edge), and the type and value of the feature are derived from observations in the UGC data from the OSN dataset. For example, we observe in the OSN dataset that a user has 57 trustors and the corresponding user node will be associated with a feature, whose type is "nTrustors" and value is 57. Table 2 lists typical sets of features used in this paper and we briefly describe them below.

Table 2. Sets of features used in the proposed model.

Feature Set	Description of Features in the Set
$\mathcal{F}_{\mathrm{PRF}}$	Statistical features for user profiles (User Profile Features)
$\mathcal{F}_{\mathrm{UGC}}$	Linguistic and stylistic features for reviews (UGC Features)
$\mathcal{F}_{\mathrm{TP}}$	Propagative features for trust propagation (TP Features)
$\mathcal{F}_{\mathrm{TAUX}}$	The first category of Auxiliary features
$\mathcal{F}_{\mathrm{TPAUX}}$	The second category of Auxiliary features

Statistical features for user profiles (User profile features)

A user's profile is the most direct depiction of the user's social capital that reflects their identity and status which in turns reflect their attitude, credibility and trustworthiness. A set of typical statistical features for user profiles built from UGC data are used in this paper. As [43] suggests, an Internet celebrity, who is in fact an active and vigorous source for disseminating information, usually have a great many followers or trustors. Thus, the *number of trustors* of a user is an obvious indicator of the evidence that how the user's being trusted by others. In the meantime, the *number of trustees* of a user also shows their engagement and importance in the online social network. The *number of reviews* posted by a user and the *number of ratings* cast by a user is a reflection of their experience, frequency and involvement in the online social network. Some online social services also have a rank system for user reviews' helpfulness, and the *numbers of a user's reviews' helpfulness being rated* as *exceptional helpful*, *very helpful*, *helpful*, *somewhat helpful* or *not helpful* are intuitive hints for the user's experience, expertise and credibility.

In the proposed model, for each user, we build the statistical features from the user's profile data and attach them to the user node; for each trust relationship, we build the statistical features from the two involving users' profiles as edge features and attach them to the corresponding edge between the user node and the trustRelation node. As explained previously, an edge between a user node and a trustRelation node may have two different types. Herein, for either type of such edges, each user profile feature has a designated type, either as a feature denoted by "u2TrU" for edges between a trustor node and a trustRelation node, or as a feature denoted by "Tr2uU" for edges between a trustRelation node and a trustee node. In this way, the model is guaranteed to distinguish the features for a user as a trustor in a trust relationship from those features for the same user acting as a trustee in another trust relationship, which conforms to our Assumption 2.

Feature vector construction. Using features listed in Table 3:

- For each user U_u, we create a feature vector $F^{\mathrm{U}}_{\mathrm{PRF}}(\mathrm{U}_u)$. Each feature of this type is a *label-observation feature*.
- For each trustor $\leftrightarrow$ trustRelation edge or trustRelation $\leftrightarrow$ trustee edge, we create a feature vector $F^{\mathrm{u2TrU}}_{\mathrm{PRF}}(\mathrm{E}_{u\leftrightarrow t})$ or $F^{\mathrm{Tr2uU}}_{\mathrm{PRF}}(\mathrm{E}_{t\leftrightarrow u})$, respectively. Each feature of this type is a *label-label-observation feature*.

Table 3. Statistical features for user profiles used in the proposed model.

Feature Name	Description
nRatings	The number of ratings a user has cast.
nRated	The number of ratings a user's reviews received.
nRated5	The number of *exceptional helpful* ratings a user's reviews received.
nRated4	The number of *very helpful* ratings a user's reviews received.
nRated3	The number of *helpful* ratings a user's reviews received.
nRated2	The number of *somewhat helpful* ratings a user's reviews received.
nRated1	The number of *not helpful* ratings a user's reviews received.
nReviews	The number of reviews posted by a user.
nTrustors	The number of trustors a user has.
nTrustees	The number of trustees a user has.

Linguistic and stylistic features for reviews (UGC features)

As previous studies [6,39,40] suggest, the linguistic characteristics and stylistic features of a review deliver the attitude, emotional status and part of expertise of the author, the quality of it implies whether or not the author is objective and unbiased, and the textual content of the review conveys the author's experience and expertise. According to Assumption 1, features extracted from reviews contribute to each user's attitude and trustworthiness, and thus affect their probability of trusting others and being trusted by others. Furthermore, according to Assumption 3, investigating this type of features also helps in finding similar users and suggesting trust relations to similar users. The linguistic and stylistic features for posts used in this paper include:

- *Parts-of-speech* (POS) used in this paper include nouns, verbs, adjectives, adverbs and conjunctions. These POS are mostly-used classes of words and may have different impacts across reviews. We use the ratio of the number of words in each POS type to the number of segments in a review as the feature value.
- The *Subjectivity* and *Polarity* of a word or a phrase describes whether the segment expresses either a *positive* or a *negative* meaning in either *strong* or *weak subjective* way. These words can have various parts-of-speech. We use the ratio of the number of these words or phrases to the number of segments in a review as the feature value.
- *Indicative* words could imply whether a post will be more credible or less convincing. They're functioning as *assertives*, *factives*, *implicatives*, *report verbs*, *hedges* or *biased words*. The lexicons are from [40,44]. Similarly, we use the ratio of the number of these words to the number of segments in a review as the feature value.

The *Affective words* (http://wndomains.fbk.eu/wnaffect.html (accessed on 1 June 2021)) and *Sentimental words* [45], which express an author's emotions, traits, sensations, attitudes or behaviors, can also be served as features for reviews. Using them may slightly increase the model's performance, however, for the sake of simplicity, we do not leverage them as features in this work.

All UGC features are attached to edges between user nodes and trustRelation nodes. Similarly as how we did with user profile features, we also mark UGC features with either type "u2TrR" or type "Tr2uR" to respect the two distinct types of edges to which they are attached.

Feature vector construction. Using features listed in Table 4:

- For each trustor ↔ trustRelation edge or trustRelation ↔ trustee edge, we create a feature vector $F_{\mathrm{UGC}}^{\mathrm{u2TrR}}(\mathrm{E}_{u\leftrightarrow t})$ or $F_{\mathrm{UGC}}^{\mathrm{Tr2uR}}(\mathrm{E}_{t\leftrightarrow u})$, respectively. Each feature of this type is a *label-label-observation feature*.

Table 4. Linguistic and stylistic features for reviews used in the proposed model.

Feature Type	Feature Name	Description: the Ratio of the Number of Specified Elements to All Segments in One of a User's Reviews
–	rPuncs	Punctuation marks
POS	rNouns	Nouns
	rAdjs	Adjectives
	rVerbs	Verbs
	rAdvs	Adverbs
	rConjs	Conjunctions
Subjectivity & Polarity	rPositives	Positive words and phrases
	rNegatives	Negative words and phrases
Indicative	rAssertives	Assertive verbs
	rFactives	Factive verbs
	rImplicatives	Implicative words and phrases
	rReports	Report verbs
	rBiases	Biased words
	rHedges	Mitigating words

Propagative features for trust propagation (TP features)

According to the trust propagation strategies described in Assumption 5, we propose two types of propagative features for trust propagation. Still take Alice, Bob and Chris for example, the proposed features are as follows.

- Direct trust propagation feature will try to capture how much Alice will trust Chris if Alice trusts Bob and Bob trusts Chris.
- Transposed trust propagation feature will try to describe how much Alice will trust Chris if Alice and Chris both trust Bob.

For any arbitrary trustRelation node T_1, we observe the state occurrence combinations of all motifs where each motif consists of three trustRelation nodes satisfying the following criteria:

- One of the three nodes in the motif is trustRelation node T_1. The other two *different* trustRelation nodes T_2 and T_3 are in the set of trustRelation nodes that are present in the dataset.
- The trustor node linked to T_1 is also linked to T_2 as a trustor node.
- The trustee node linked to T_1 is also linked to T_3 as
  - either a trustee node (for direct trust for trustRelation node T_1) while the trustee node of T_2 is the trustor node of T_3,
  - or a trustor node (for transposed trust for trustRelation node T_1) while the trustee node of T_2 is also the trustee node of T_3.

In other words, for each motif, the observing target including the motif itself consists of a trustor node, a trustee node, their corresponding trustRelation node, a third user in whose trust relationships either the trustor/trustee or trustor/trustor are shared, and their corresponding trustRelation nodes. Figure 4 illustrates the topological diagram for direct trust and transposed trust in the proposed model.

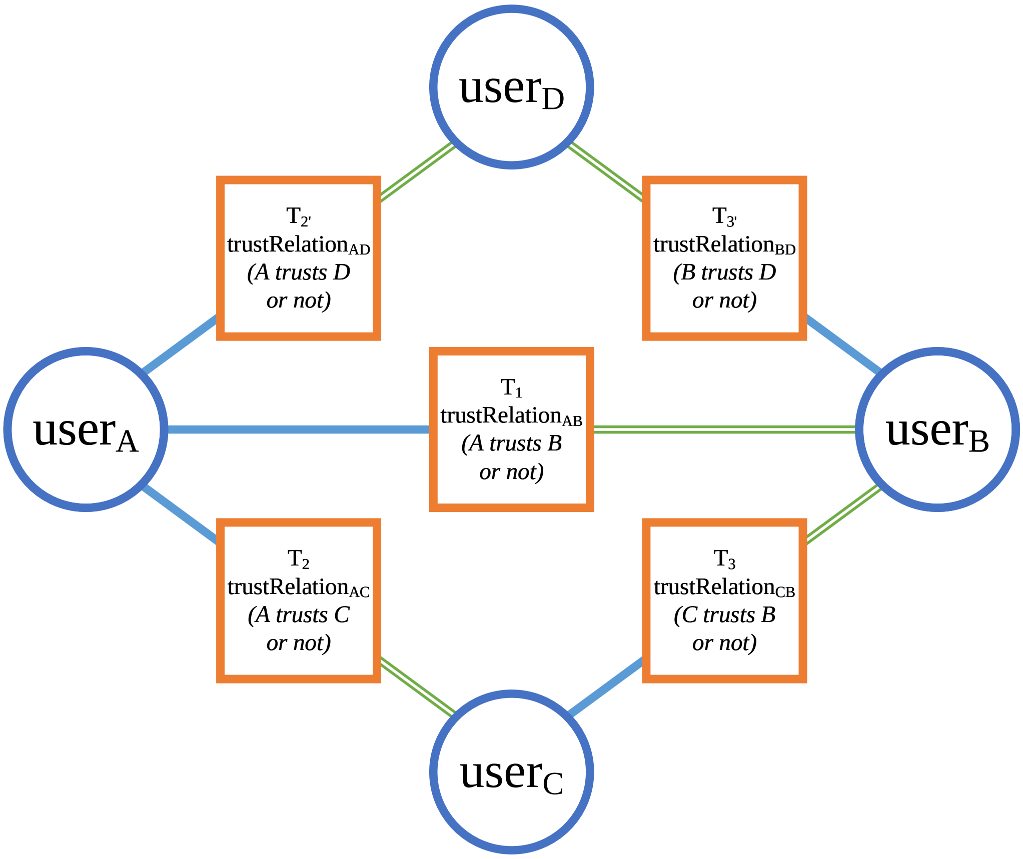


Figure 4. Illustration of direct trust and transposed trust for a trustRelation node in a 3-trustRelation-nodes motif. (T$_2$ and T$_3$ contribute direct trust to T$_1$; and T$_{2'}$ and T$_{3'}$ contribute transposed trust to T$_1$).

We build a trustRelation node's trust propagation features based on the numbers of state sequence occurrences of the three trustRelation nodes in each motif that the node has. Use the notation from the above criteria, for a trustRelation node T$_1$ and all its possible 3-trustRelation-nodes motifs that meet the criteria, a total of 16 trust propagation features are built, through the eight state sequences of the three trustRelation nodes in a motif in the specific order of T$_1$, T$_2$ and T$_3$. The feature values are either 1, if at least one motif with a corresponding state sequence exists, or 0, otherwise. Note that only trustRelation nodes representing trust or no trust relationships that exist in the observed dataset will get accounted for generating trust propagation feature values. As listed in Table 5, eight of the 16 features in this type are for direct trust propagation and the other eight are for transposed trust propagation.

Feature vector construction. Trust propagation features of each type will be generated as node features and get attached to trustRelation nodes. Using features listed in Table 5:

- For each trustRelation node T$_t$, we check if any instance of the 16 state sequences exists to generate trust propagation features, by applying the above criteria to all 3-trustRelation-nodes motifs in which T$_t$ acts as T$_1$, and then create a feature vector $F_{\mathrm{TP}}^{\mathrm{T}}(\mathrm{T}_t)$ to include these features. Each of them is a *label-observation feature*.

Table 5. Trust propagation features used in the proposed model.

Feature Type	Feature Name	Sequenced "Labels" of Nodes in Motif		
		$\mathbf{T}_{A\to B}$ / $\mathbf{T}_1$	$\mathbf{T}_{A\to C}$ / $\mathbf{T}_2$	$\mathbf{T}_{C\to B}$ / $\mathbf{T}_3$
Direct Trust	d000	N	N	N
	d001	N	N	Y
	d010	N	Y	N
	d011	N	Y	Y
	d100	Y	N	N
	d101	Y	N	Y
	d110	Y	Y	N
	d111	Y	Y	Y
		$T_{A\to B}$ / T_1	$T_{A\to D}$ / T_2	$T_{B\to D}$ / T_3
Transposed Trust	t000	N	N	N
	t001	N	N	Y
	t010	N	Y	N
	t011	N	Y	Y
	t100	Y	N	N
	t101	Y	N	Y
	t110	Y	Y	N
	t111	Y	Y	Y

Refer to Figure 4 for $T_{A\to B}$, $T_{A\to C}$, $T_{C\to B}$, and $T_{B\to D}$. For a trustRelation node: "Y" indicates an existing trust and "N" a nonexistent one.

Auxiliary features

For better modeling real-world correlations among users and trust relationships and for the proposed method to work properly in the model's probabilistic inference, certain auxiliary edge features are built and attached to the model. They are called auxiliary in this paper because the proposed model and the trust inference approach will still work without adding them, though inefficiently.

We build the auxiliary edge features through the inspiration of the *Ising model* [46] (or more generally the *Potts model*) in statistical mechanics. These models imply that two directly connected nodes of the same type tend to be in the same state. This inspires us that the state–state pair of two directly connected nodes of different types might also follow certain statistical rules. In this paper, two categories of auxiliary edge features are proposed as follows.

1. One category of auxiliary edge features will be attached to each edge between a user node and a trustRelation node. Their labelnames are, respectively, prefixed with "u2TrT" and "Tr2uT" for features on a *trustor–trustRelation edge* and features on a *trustRelation–trustee edge*. This setting matches the construction of our probabilistic graphical model where edges between user nodes and trustRelation nodes have different types. Such an setting allows the model to distinguish how differently a trustor or a trustee affects a trust relationship's formation.
   Feature vector construction. For each trustor ↔ trustRelation edge or trustRelation ↔ trustee edge, we create a feature vector $F^{\mathrm{u2TrT}}_{\mathrm{TAUX}}(\mathrm{E}_{u\leftrightarrow t})$ or $F^{\mathrm{Tr2uT}}_{\mathrm{TAUX}}(\mathrm{E}_{t\leftrightarrow u})$, respectively. Each feature in this category is a *label-label feature*.
2. The other category of auxiliary edge features will be attached to edges between trustRelation nodes that are involved in the motif structure explained previously. Similarly to trust propagation features, features in this category follow the concept of propagative trust, i.e., direct trust propagation and transposed trust propagation, and grant values of each of them with either 0 for direct trust or 1 for transposed trust. However, different from the trust propagation features which are node features, they are edge features trying to "filter out similarly behaving trustRelation nodes".

Feature vector construction. For each trustRelation $\leftrightarrow$ trustRelation edge, we create a feature vector $F^{\mathrm{T}}_{\mathrm{TPAUX}}(\mathrm{E}_{t\leftrightarrow t'})$. Each feature in this category is a *label-label-observation feature*.

The current implementation of the proposed model does not support inference on cliques yet, which will be discussed below. And thus, it hinders the employment of the second category of auxiliary features (*label-label-observation features*) on edges between trustRelation nodes, which require the prerequisite of the existence of trustRelation–trustRelation edges. However, we use the same inference method for pairwise graph structure approximately for cliques, and we'd like to leave the inference on cliques as future work to complete.

3.2.3. Model Formulation

The random variables in our model are the states or labels of corresponding nodes. For a user node, its state is a predefined measurement for the user from the original online social network. It can be direct statistics of this user or carefully handcrafted measurement calculated from information obtained from their OSN data. For demonstration purpose and brevity, we use user categories defined by the online social service as user node states in this paper. For a trustRelation node, which represents the trust relationship between the trustor user and the trustee user, the random variable's state is the trust relationship's existence in the OSN dataset. As each node has a state, each edge has a state–state pair (or a transition state) that is determined by the states of the two nodes connected by it. Table 6 summarizes state configurations used in this paper.

Table 6. Configurations of node state and edge state–state pair.

Type	States	Description
Node		
user (u)	0, 1, 2	User categories defined by the OSN.
trustRelation (t)	0	Such an relationship is observed.
	1	Such an relationship is un-observed.
Edge (U: state of user node, T: state of trustRelation node)		
$\mathrm{E}_{u\leftrightarrow t}$	UT: 00, 01, 10, 11, 20, 21	state–state pair consisting of U and T.
$\mathrm{E}_{t\leftrightarrow u}$	TU: 00, 01, 02, 10, 11, 12	state–state pair consisting of T and U.
$\mathrm{E}_{t\leftrightarrow t'}$	TT: 00, 01, 10, 11	state–state pair consisting of T and T.

Due to the way that we model trust relationships as random variable nodes in the probabilistic graph, the smallest significant structure in the resulting graph is pairwise. (Although there will be smallest cliques that consist of three trustRelation nodes if the second category of auxiliary features is leveraged, we still model the conditional random field and run probabilistic inference on it pairwisely.) Different pairwise structures are connected via the common user nodes or trustRelation nodes. It is worth noting that each node, if connected by a dummy node, can also be viewed as a pairwise structure, and thus we call each node a unitary structure.

Each unitary structure has a set of associated *node feature functions* and each pairwise structure has a set of *edge feature functions*. In our problem setting, we attach features to each node and edge. The features used in this paper are the user profile features, UGC features, and trust propagation features.

We use T and T^* to denote the set of all known trust relationships and the set of trust links whose states are unknown, respectively. Let Ψ denote the set of feature weights, and the proposed model that computes a conditional distribution can be defined as

$$P(\mathrm{T}^*|\mathrm{U},\mathrm{T};\Psi) = \frac{1}{Z(\mathrm{U},\mathrm{T})}\prod_i \varphi_i(\mathrm{U},\mathrm{T};\Psi), \tag{4}$$

where $Z(\cdot)$ is the normalization constant, and φ_i is the i_{th} potential function for either a unitary structure or a pairwise structure.

For any node $\mathrm{V}_i \in (\mathrm{U},\mathrm{T},\mathrm{T}^*)$ or any edge $\mathrm{E}_{j:\langle \mathrm{V}_s,\mathrm{V}_t\rangle}$ connecting nodes V_s and V_t, the potential functions for either unitary structures or pairwise structures used in this paper are, respectively, defined in the log-linear form as follows.

$$\varphi_i(\mathrm{V}_i;\Psi_{\mathrm{V}}) = \exp\left(\sum_k \psi_k f_{ik}(\mathrm{V}_i)\right), \tag{5}$$

$$\varphi_j(\mathrm{E}_{j:\langle \mathrm{V}_s,\mathrm{V}_t\rangle};\Psi_{\mathrm{E}}) = \exp\left(\sum_k \psi_k f_{jk}(\mathrm{E}_{j:\langle \mathrm{V}_s,\mathrm{V}_t\rangle})\right), \tag{6}$$

where $f_{ik}(\cdot)$ is the k_{th} feature function in the i_{th} structure. In the way that features are weighted and then linearly aggregated, the interplay among features of a same node or edge is collectively numericalized.

3.3. Probabilistic Model Inference and Interpretation

From the above-mentioned model construction, we know that there will be loops in the probabilistic graph if bidirectional trust relations exist. That means if Alice and Bob trust each other, which is common in the real world, then the graph containing only the two users and their trust relationships is no longer a linear-chain or a tree, as shown in Figure 3.

For the probabilistic model inference, collectively predicting all needed trust relationships from the online social network involves iterating through an exponential number of possible label combinations, and thus requires exponential time. Furthermore, as the probabilistic graph contains loops, and exact inference on such a general graphical model is thus intractable, approximations are employed. We propose to solve the inference problem using the *loopy belief propagation* (LBP) technique. As a *belief propagation* (BP) method variant, the LBP is a *message passing* algorithm but requires a slightly different schedule of message updating rules from the vanilla BP method.

It is worth noting that each pairwise structure in the probabilistic graph only connects two random variable nodes, and a unitary structure can be viewed as if there is a dummy node linked to it so that a pseudo-pairwise structure exists. Therefore, we can safely skip the factor graph framework and send messages directly between each pair of nodes connected by an edge. This way of handling message passing is equivalent to the original BP algorithm.

We chose to update messages synchronously, i.e., in each time epoch, each pair of nodes exchange messages, if they are connected by an edge. In the iterative message updates, each node's belief is normalized so that the normalized belief approximates the node's marginal probability and is further considered as the new local evidence (also called compatibility or potential) at this node. Similarly, for an edge and the two nodes connected by it, we use the normalized belief of the pairwise structure as its local evidence. In the model inference, passing messages, calculating node/edge beliefs, and updating messages are repeated until message convergence or an allowed maximum number of iterations is reached.

When the LBP is done, we normalize each node's belief so that it approximates the node's marginal distribution. Through the marginal probabilities of all possible labels for a selected trustRelation node, the trust relationship indicated by this node can be determined by Equation (1). Such probabilities for the trustRelation node can be further interpreted from the message passing perspective.

Let m_{ji} denote a message sent from node i's neighboring node j to it, x_i be the random variable at node i, and $\varphi_i(x_i)$ and $\varphi_{ij}(x_i, x_j)$ be the local evidence at node i (unitary structure) and edge $\langle i, j\rangle$ (and its connected nodes i and i, pairwise structure). According to BP, the belief $b_i(x_i)$ at a node i before normalization, which will then approximate the variable's marginal probability, is proportional to the product of the local evidence at this node and all the messages coming to it:

$$b_i(x_i) \propto \varphi_i(x_i) \prod_{j \in N(i)} m_{ji}(x_i), \tag{7}$$

where $N(i)$ denotes the set of nodes directly neighboring node i. The messages are determined by message update rules as follows,

$$m_{ji}(x_i) \leftarrow \sum_{x_j} \varphi_j(x_j)\varphi_{ji}(x_j, x_i) \prod_{k \in N(j)\backslash i} m_{kj}(x_j). \tag{8}$$

Analogously to Equation (7), the pairwise belief $b_{ij}(x_{ij})$ at a pairwise structure will be

$$b_{ij}(x_{ij}) \propto \varphi_{ij}(x_i, x_j)\varphi_i(x_i)\varphi_j(x_j) \prod_{k \in N(i)\backslash j} m_{ki}(x_i) \prod_{k \in N(j)\backslash i} m_{kj}(y_j), \tag{9}$$

Without loss of generality, we take Figure 5 as an example for the following discussion. $\mathrm{U_A}$, $\mathrm{U_B}$, $\mathrm{T_{AB}}$ are the random variables of the trustor node A, the trustee node B and the trustRelation node linked to them, respectively. Since the trustRelation node is linked to a trustor node and a trustee node by only two edges, with Equation (7), the belief at it is proportional to

$$b_{\mathrm{T_{AB}}}(\mathrm{T_{AB}}) \propto \varphi_{\mathrm{T_{AB}}}(\mathrm{T_{AB}}) \prod_{i \in N(\mathrm{T_{AB}})} m_{i \to \mathrm{T_{AB}}}(\mathrm{T_{AB}}), \tag{10}$$

and the two messages (M_A and M_B in the figure) sent to the trustRelation node are, respectively

$$m_{\mathrm{U_A} \to \mathrm{T_{AB}}}(\mathrm{T_{AB}}) \leftarrow \sum_{\mathrm{U_A}} \varphi_{\mathrm{U_A}}(\mathrm{U_A})\varphi_{\mathrm{U_A},\mathrm{T_{AB}}}(\mathrm{U_A}, \mathrm{T_{AB}}) \prod_{k \in N(\mathrm{U_A})\backslash \mathrm{T_{AB}}} m_{k \to \mathrm{U_A}}(\mathrm{U_A}), \tag{11}$$

$$m_{\mathrm{U_B} \to \mathrm{T_{AB}}}(\mathrm{T_{AB}}) \leftarrow \sum_{\mathrm{U_B}} \varphi_{\mathrm{U_B}}(\mathrm{U_B})\varphi_{\mathrm{U_B},\mathrm{T_{AB}}}(\mathrm{U_B}, \mathrm{T_{AB}}) \prod_{k \in N(\mathrm{U_B})\backslash \mathrm{T_{AB}}} m_{k \to \mathrm{U_B}}(\mathrm{U_B}). \tag{12}$$

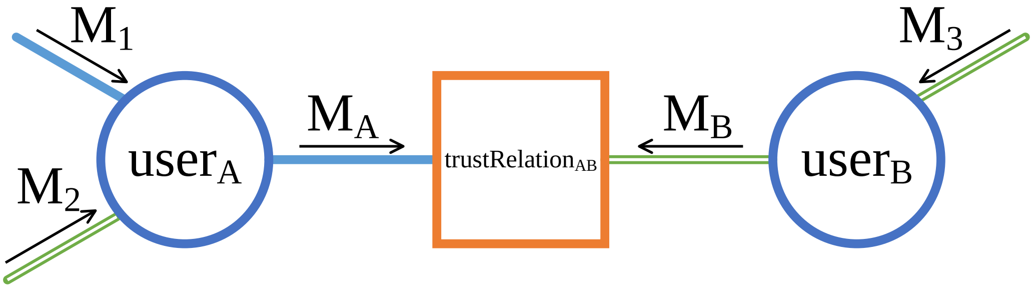


Figure 5. An illustration of passing messages through a trustor node $\mathrm{U_A}$ and a trustee node $\mathrm{U_B}$ to a trustRelation node $\mathrm{T_{AB}}$.

Let $\Lambda_x(y)$ be the multiplication of the local evident at node x and edge $\langle x, y\rangle$, and substitute Equations (11) and (12) and the messages demonstrated in the figure into Equation (10). The belief at the trustRelation node amounts to

$$b_{\mathrm{T_{AB}}}(\mathrm{T_{AB}}) = \kappa\varphi_{\mathrm{T_{AB}}}(\mathrm{T_{AB}})(M_A M_B) \tag{13}$$

$$= \kappa\varphi_{\mathrm{T_{AB}}}(\mathrm{T_{AB}})\left[\sum_{\mathrm{U_A}}\Lambda_{\mathrm{U_A}}(\mathrm{T_{AB}})(M_1 M_2)\right]\left[\sum_{\mathrm{U_B}}\Lambda_{\mathrm{U_B}}(\mathrm{T_{AB}})M_3\right], \tag{14}$$

where κ is a normalization constant.

It follows that the trustRelation node's belief is proportional to all local evidence at this node and two messages sent by the trustor node and the trustee node. The local evidence is the calculated potential through the compatibility function (energy function, refer to Equations (5) and (6). Based on the trustor's attitude, experience, awareness of other users' expertise and trustworthiness, etc., collected by Equation (11), the message M_A sent by the trustor node tells the trustRelation node how much the trustor believes they will issue or not issue a trust relationship to the trustee. Likewise, through the trustee's expertise, experience and evidential suggestions from other users' trust, etc., constituted by Equation (12), the message M_B sent from the trustee node tells the trustRelation node how much the trustee thinks their behaviors should be recognized by the potential trustor so that the trustor will trust them or not.

One can conclude that both the trustor and the trustee are involved in forming a trust relationship, obviously, in a real-world social network, and they have different contributions to the relationship formation, which is in accordance with Assumption 2, which we made previously.

3.4. Parameter Estimation

For the model to be able to infer trust, we need to know each parameter in the model, i.e., the weights for all features. In this section, we embrace the simple but effective gradient descent method to minimize the model's negative likelihood so that the training data will achieve locally highest probability under the model.

From Equations (4)–(6), the conditional log-likelihood with respect to the set of feature weights Ψ to be maximized can be obtained as follows.

$$l(\Psi) = -\log Z(\cdot) + \sum_i \log\left[\varphi_i(\mathrm{U}, \mathrm{T}; \Psi)\right] \tag{15}$$

$$= -\log Z(\cdot) + \sum_i\left(\sum_k \psi_k f_{ik}(\mathrm{V}_i)\right) + \sum_j\left(\sum_k \psi_k f_{jk}(\mathrm{E}_{j:\langle \mathrm{V}_s, \mathrm{V}_t\rangle})\right), \tag{16}$$

and with the LBP algorithm, the approximated gradients of the likelihood (the partial derivatives with respect to feature ψ_k) are

$$\frac{\partial l}{\partial \psi_k} = \sum_i\sum_k f_{ik}(X_i) - \sum_i\sum_k f_{ik}(X_i)q(X_i) + \frac{\psi_k}{\sigma^2}, \tag{17}$$

where X_i can be either a unitary or a pairwise structure; $q(\cdot)$ is the approximated marginal, i.e., the belief of the unitary or pairwise structure, which can be determined by running a pass of the LBP algorithm on the graph; and the last term is the regularization term to prevent over-fitting.

As was discussed in the previous section, calculating the logarithm of the normalizing constant $\log Z(\cdot)$ is intractable, and thus we use Bethe Energy [47] to further approximate it:

$$\begin{aligned}\log Z(\cdot) &\approx l_{\mathrm{BETHE}}(\psi, q) \\ &= -\textstyle\sum_{s,t}\sum_{\mathrm{V}_s,\mathrm{V}_t} q(\mathrm{V}_s,\mathrm{V}_t)[\log q(\mathrm{V}_s,\mathrm{V}_t) - \log P(\mathrm{V}_s,\mathrm{V}_t)] \\ &\quad + \textstyle\sum_s\sum_{\mathrm{V}_s}[d(\mathrm{V}_s) - 1]q(\mathrm{V}_s)[\log q(\mathrm{V}_s) - \log P(\mathrm{V}_s)]\end{aligned} \tag{18}$$

where $d(\mathrm{V}_s)$ is the degree of node V_s; $P(\mathrm{V}_s)$ and $P(\mathrm{V}_s, \mathrm{V}_t)$ are, respectively, the initial potentials of their unitary and pairwise structures; and $q(\cdot)$ and $q(\cdot, \cdot)$ are corresponding optimal marginal distributions of approximated beliefs through LBP.

Between gradient descent epochs, the new feature weight vector $\psi^{(m)}$ is computed from the old vector $\psi^{(m-1)}$ by

$$\psi^{(m)} = \psi^{(m-1)} + \alpha_m \nabla l(\psi^{(m-1)}), \tag{19}$$

where $\nabla l(\cdot)$ is the partial derivatives calculated through Equation (17), and $\alpha_m > 0$ is a step size controlling the distance the parameter moves in the direction of the gradient, which is also called the learning rate.

3.5. Implementation

Our implementation of the proposed model, including the model construction, inference, and relevant training and predicting procedures, was based on the framework introduced in our previous work [48]. For the sake of brevity, we only discuss some concerns on the model implementation in this section which may greatly affect the model's performance.

Numerical overflows and underflows. These are very common in BP and other message passing algorithms. For example, some terms in the potential functions of unitary or pairwise structures (e.g., Equations (5) or (6)) may be overflowed due to the exponential calculations; in message passing (e.g., Equation (8)), if the messages passed via each edge are not constrained, then some messages will exhibit underflow after certain iterations of message updates; and calculating beliefs (e.g., Equation (7) and (9)) may cause overflow or underflow or both. The trick to tackling the overflow problem here is to shift every exponent in the calculated potential by subtracting the largest exponent. For the underflow issue during message passing, it is necessary to normalize messages frequently.

Parameter learning rate in model training. According to Equation (19), if the learning rate α_m for step m is too large, the new parameters will move too far in the direction of the gradient; if it is too small, the training procedure will be very slow to accomplish. Thus, it is essential to properly schedule learning rates in the whole model training process. A simple common approach is to let α_m decrease slowly to 0 as step m grows, which is

$$\alpha_m = \frac{1}{\sigma^2(\alpha_0 + m - 1)}, \tag{20}$$

where α_0 is the initial learning rate and σ^2 is the L2 regularization. The initial learning rate α_0 can be manually set or obtained by running a few passes of gradient descent over the graph [49].

Implementation of bi-directional message passing. For bi-directionally passing messages via each edge, the implementation of a non-directional edge in the model is built as two mutually opposite directional edges. Consequently, the number of *real functioning edges* in the implementation will be doubled and the number of calculations in the probabilistic inference procedure will greatly increase. In order for the model to perform LBP efficiently, parallelism could be deployed in synchronous message passing. It is also beneficial to offload the LBP algorithm to *graphics processing unit* (GPU) devices to further accelerate the model inference and training.

4. Experiments

We performed experiments with data (the dataset is publicly accessible via https://www.cse.msu.edu/~tangjili/trust.html (accessed on 10 June 2017)) from a typical online social network, Ciao, where users are allowed to build trust network, cast ratings, post reviews on a variety types of items, and rate other's reviews. We chose Ciao because of the availability of a relatively full web of trust and abundant user-generated content for explicit and inexplicit trust for trust inference.

4.1. Data

For fast model evaluation and comparison with other binary classifiers, we extracted a portion from the full data as our dataset used for experiments. As user links in social networks are often sparse, it is obvious that the observed trust relationships takes up only a very small fraction, whereas unobserved ones are common, which means the data are unbalanced. To deal with the imbalance in the dataset and for the sake of simplicity, we undersampled unobserved relationships. The statistics for the dataset used in experiments are listed in Table 7.

Table 7. Dataset specification.

Number of users	14,317	
Number of reviews	24,406	
Number of reviews per user	1.7	
Number of trust relationships	Y: 87,804	N: 78,863
Web of trust density	Y: 0.00043	N: 0.00038

For a trust relationship: "Y" indicates an existing trust and "N" a nonexistent one.

Features introduced in Section 3.2.2, including contextual features and relational features, were constructed with information extracted from the dataset. For reviews, we deployed an annotator by CoreNLP (https://stanfordnlp.github.io/CoreNLP/ (accessed on 20 June 2017)) to extract words and phrases from texts and build stylistic and linguistic features. Detailed feature statistics are listed in Supplementary Materials.

Although we carried out the experiments on only one dataset, the proposed method is universally applicable to various online social networks. With only a set of users and the set of trust relationships from other data sources, the model can still be built successfully and then infer trust, though without adequate relevant features the inference may perform unsatisfactorily. If additional features are available, the proposed method will prove its efficiency in trust relationship prediction.

4.2. Experimental Settings

4.2.1. Comparison Methods

We chose several easy-to-implement state-of-the-art binary classifiers as baseline comparison methods, including *support vector machine* (SVM) with a radial basis function (RBF) kernel, *decision tree* (DT), and *random forest* (RF). Generally speaking, linear SVMs are interpretable but less efficient than SVMs with an RBF kernel that are partially interpretable; DTs provides interpretable results, and RFs deliver better results than DTs do but decrease interpretability.

We did not compare our model with the methods proposed in [33–35], as ours is supervised learning. We also did not conduct comparison experiments between ours and models based on neural networks. Although these models probably could, and most likely would, achieve better performances than ours, to interpret these models is still a problem.

4.2.2. Evaluation Metrics

As we construct our mission of trust relationship inference as a binary classification task in this paper, we used the common metrics for classification evaluation. In detail, we used *Accuracy*, *Precision*, *Recall*, and F_1 *score* for evaluations.

For the set of user trust relationships that are to be inferred by a model or a method, we denote the number of all elements in the set by C; the number of observed trust relationships that were also predicted to be existent (true positive data points) by tp; and the numbers of false negative, false positive, and true negative ones by fn, fp, and tn, respectively. Then, the Accuracy is defined as

$$\mathrm{Accuracy} = \frac{\mathrm{tp} + \mathrm{tn}}{\mathrm{C}}, \tag{21}$$

and Precision, Recall, and F_1 score are, respectively, defined as

$$\text{Precision} = \frac{\text{tp}}{\text{tp} + \text{fp}}, \tag{22}$$

$$\text{Recall} = \frac{\text{tp}}{\text{tp} + \text{fn}}, \tag{23}$$

$$\text{F}_1 = \frac{2\text{tp}}{2\text{tp} + \text{fp} + \text{fn}}. \tag{24}$$

From the metric definitions, the higher one metric is, the better performance a model or a method has.

4.2.3. Experiment Setup

In this paper, two sets of experiments were conducted.

1. For model validation and comparisons, we conducted experiments using the proposed model and comparison methods with different feature set combinations on the split training and test datasets, and then compared the resulting performances with the evaluation metrics.
2. For privacy-restrict online social network analysis, experiments were carried out with partially reduced data to further explore the proposed model's trust inference capability in a real-world scenario. Hereinafter, the reduced data means that features from a certain set for a portion of users were missing for a specific experiment. As stated earlier in Section 1, in real-world online social networks, some users may choose to opt out of part of or all of their data being used by online social services.

As was discussed in Section 3.2.2, the proposed model will still work without auxiliary features. Nevertheless, the types of auxiliary features are a particular coexistent by-product of how the proposed model was built, and also reflect the real-world mechanism of trust relationship formation. Therefore, these features were used in all experiments for the proposed model, acting as part of the foundation of the model. Note that these features only work with the proposed model.

In all experiments for model validation and comparison, we used different combinations of feature sets to show each method's ability in trust inference. Table 8 lists the different feature sets for experiments. (Refer to Table 2 for the types of features used in this paper.) For the comparison methods, the feature set combinations available for them are similar but without any auxiliary features from $\mathcal{F}_{\text{TAUX}}$ or $\mathcal{F}_{\text{TPAUX}}$.

For model learning and predicting, the set of user trust relationships was randomly split into a training set and test set, and the ratios of sizes of the two sets were 50–50%, 0–40%, 70–30%, 80–20%, and 90–10%. (A better way to split the data is to split data according to trust relationship creation time. However, in the chosen dataset, such information of trust relationship creation time was missing, and thus we resorted to splitting the data randomly.) All the second set of experiments used the 90–10% split for training and test datasets.

For the second set of experiments, the reduced feature data were generated by randomly removing one type of features from a certain percentage of users. The percentages for feature removal are 20%, 40%, 60% and 80%, respectively. If two types of features are used at the same time for an experiment, the random removal for each type of features is independent.

Other experiment setup included: in all experiments for the proposed model, we set the maximum number of gradient descent epochs to 20 and the maximum number of LBP iterations to 10.

Table 8. Sets of features used for experiments.

# of Experiment Set		# of Experiment	Feature Set Contents			
1st	2nd		$\mathcal{F}_{\text{PRF}}$	$\mathcal{F}_{\text{UGC}}$	$\mathcal{F}_{\text{TAUX}}$	$\mathcal{F}_{\text{TP}} + \mathcal{F}_{\text{TPAUX}}$
✓		1			✓	
✓	✓	2	✓		✓	
✓	✓	3		✓	✓	
✓	✓	4			✓	✓
✓	✓	5	✓	✓	✓	
✓	✓	6	✓		✓	✓
✓	✓	7		✓	✓	✓
✓		8	✓	✓	✓	✓

4.3. Results and Discussion

4.3.1. The First Set of Experiments with All Possibly Usable Feature Data

Performance of the proposed method with different feature sets

Firstly, we report the trust inference performance of the proposed method. The experiments were conducted with eight different feature set combinations on the five split training and test datasets, and accuracy and $\mathbf{F_1}$ score are reported for each of them. As shown in Figure 6, the results are organized into three groups by how the feature sets were composed: the first group contains results from experiments 1, 2, 5, 6, and 8, which illustrates how integrating the user profile feature set $\mathcal{F}_{\text{PRF}}$ into the model affects the model's performance; similarly, the second group contains experimental results from experiments 1, 3, 5, 7, and 8 to show the power of using UGC feature set $\mathcal{F}_{\text{UGC}}$; the third group of 1, 4, 6, 7, and 8 depicts the effect of trust propagation feature set $\mathcal{F}_{\text{TP}} + \mathcal{F}_{\text{TPAUX}}$.

With either a single feature set or a combination of feature sets, the proposed method outperformed three naive classifiers, *uniformly random guess*, *random selected class*, and *majority class*, which, respectively, achieved accuracies of 0.5000, 0.5014, and 0.5269 according to the dataset specification (refer to Table 7). From the reported results evaluated by the accuracy metric, one can conclude that the proposed method is efficient and capable of inferring trust relationships.

As was previously discussed, with only the first category of auxiliary features, the proposed method will still worked, and it also achieved above-average performance. They are special features that only work with our model but not for other comparison methods. This makes our model with this set of features promising as a supervised learning model, when abundant data, such as user-generated content, are available as data input.

A quick glance over the performance results regarding both accuracy and F_1 score shows that substituting additional features into the model does improve the model's performance. However, the results differ a bit when the added combination of additional feature sets varies. The observations are as follows.

1. On top of the first category of auxiliary features ($\mathcal{F}_{\text{TAUX}}$), adding a single feature set ($\mathcal{F}_{\text{PRF}}$, $\mathcal{F}_{\text{UGC}}$, or $\mathcal{F}_{\text{TP}} + \mathcal{F}_{\text{TPAUX}}$) into the model will improve the model's performance. The use of the UGC feature set $\mathcal{F}_{\text{UGC}}$ improves the model's accuracy (and F_1 score) greatly by 42.45% to 50.22% (27.80% to 30.76%), followed by the user profile feature set $\mathcal{F}_{\text{PRF}}$ by 14.92% to 20.87% (1.16% to 6.60%), and then the trust propagation feature set $\mathcal{F}_{\text{TP}} + \mathcal{F}_{\text{TPAUX}}$ by 0.35% to 1.01% (0.34% to 1.32%).
2. Although adding a single feature set $\mathcal{F}_{\text{TP}} + \mathcal{F}_{\text{TPAUX}}$ did not greatly improve the performance, it could help another single feature set $\mathcal{F}_{\text{PRF}}$ to achieve better results. This can be seen from experiment 6 and experiment 2, where the model performance was improved by 11.26% to 14.27% in accuracy (Figure 6a) and 13.60% to 18.09% in F_1 score (Figure 6d).
3. Using all types of features (in experiment 8) does not always promise the best result. The performance for the proposed model with such feature sets was close to the

performance of the model with the UGC feature set $\mathcal{F}_{\text{UGC}}$ with or without other feature sets.

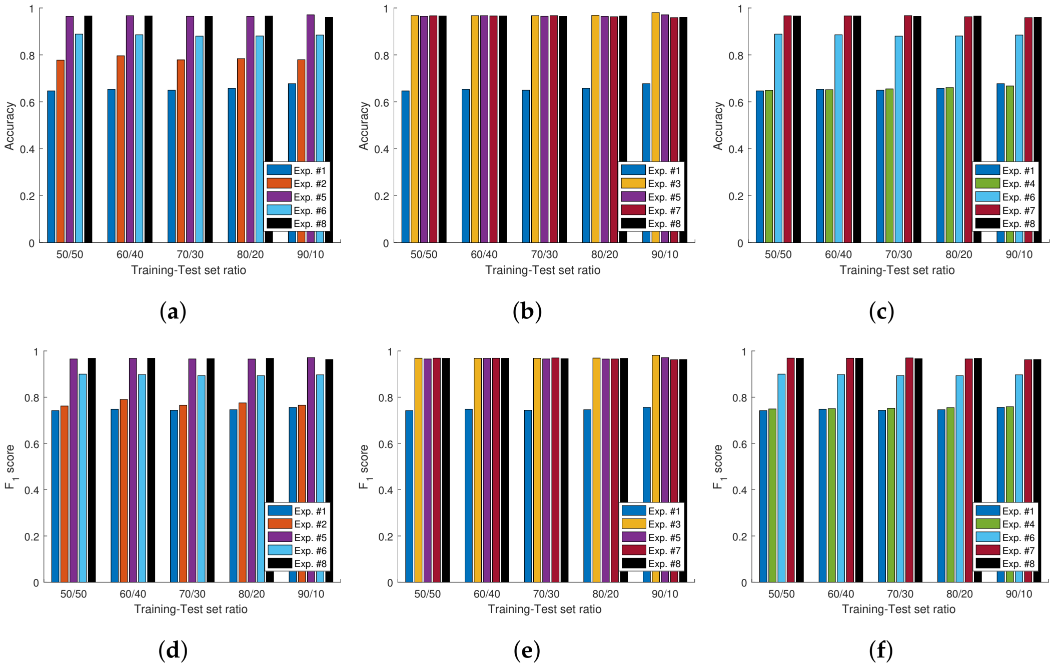


Figure 6. Performance results of the proposed model evaluated by Accuracy (Acc) and F$_1$ score for the first set of experiments. (**a**) Experiments 1, 2, 5, 6, 8 (Acc). (**b**) Experiments 1, 3, 5, 7, 8 (Acc). (**c**) Experiments 1, 4, 6, 7, 8 (Acc). (**d**) Experiments 1, 2, 5, 6, 8 (F$_1$). (**e**) Experiments 1, 3, 5, 7, 8 (F$_1$). (**f**) Experiments 1, 4, 6, 7, 8 (F$_1$).

It is expected to see the model get better results when additional features are added into it. It is also foreseeable that using UGC features ($\mathcal{F}_{\text{UGC}}$) should work better than using other types of features, as UGC data are usually more abundant than other types of features, both from a dataset and in the real world. The expected result will in turn validate Assumption 1, that a user's generated contents hold the representation of the user's trust information. All the good performance results came from a rich amount of UGC data, and so, a legitimate question arises, "What if some of the UGC data are missing possibly due to any privacy-related constraints?" Or more precisely, "What would the model performance become if some of the UGC data are not available?" The second set of experiments conducted with reduced feature data will shed light on the answers to them.

As for the trust propagation features, apart from the data imbalance, there is another possible reason that they do not always improve the model's performance: the probabilistic inference is performed on pairwise structures rather than cliques. Nevertheless, the trust propagation features do improve the model's performance when working with the statistical user profile features.

Performance Comparisons

Secondly, we report the comparative performance results achieved by using our model and other methods discussed in the previous section. For these experiments, all available feature datasets and their combinations are free to use for experimenting. The performances for each method evaluated by the accuracy and F$_1$ score metrics, and experiments in which the best performances were achieved are reported in Tables 9 and 10. Full results, including precision and recall for each experiment, are reported in the Supplementary Materials. As has been already stated, the auxiliary features only fit in the proposed model, and all the other types of features can be used for all methods.

Table 9. First experiment set: best performance comparison for different methods by Accuracy.

Training–Test	Our Model	SVM	DT	RF
50–50%	0.9678 (#3)	0.9106 (#6)	0.8778 (#6)	0.9212 (#8)
60–40%	0.9673 (#3)	0.9139 (#6)	0.8847 (#6)	0.9239 (#8)
70–30%	0.9675 (#7)	0.9120 (#6)	0.8853 (#6)	0.9215 (#8)
80–20%	0.9684 (#3)	0.9093 (#6)	0.8729 (#6)	0.9152 (#8)
90–10%	0.9804 (#3)	0.9143 (#6)	0.8741 (#6)	0.9198 (#8)

Table 10. First experiment set: best performance comparison for different methods by F_1 score.

Training–Test	Our Model	SVM	DT	RF
50–50%	0.9688 (#7)	0.9160 (#6)	0.8850 (#6)	0.9247 (#8)
60–40%	0.9684 (#7)	0.9190 (#6)	0.8923 (#6)	0.9275 (#8)
70–30%	0.9698 (#7)	0.9170 (#6)	0.8925 (#6)	0.9250 (#8)
80–20%	0.9691 (#3)	0.9141 (#6)	0.8793 (#6)	0.9188 (#8)
90–10%	0.9810 (#3)	0.9193 (#6)	0.8812 (#6)	0.9238 (#8)

From the experimental results listed in the tables and the Supplementary Materials, the first observation is that our proposed method outperforms other comparison methods in terms of accuracy and F_1 score, if all types of features are free to use.

The second observation is that UGC features ($\mathcal{F}_{\text{UGC}}$) were always involved in the used features when all methods achieved their respective best or second best performances. The fact that there is trust construct embedded in UGC data is proved again, but through the results by comparison methods this time. Consequently, one can harvest a user's trust information from UGC data from the user and his "friends" including trustors and trustees, which thus bears witness to Assumption 1 we made in Section 3.1.

Last but not least, in contrast to the results achieved by our proposed model, SVM and decision tree generated worse results by adding UGC features than user profile features, and all three comparison methods had improved results by using trust propagation features. One possible reason for the performance divergence's cause is the very large number and dimensions of UGC features employed in the three comparison methods; meanwhile, a few yet powerful trust propagation features greatly improved their performance when working with either UGC features or user profile features. Another reason that our model behaved differently in these experiments is that ours can balance a massive number of features which may counter each other, and can also "capture and numericalize" interplayed features.

A peek at Recall: higher Precision or higher Recall for trust relationship prediction in OSNs?

For a classification task, the outcome for it determines the expectations for the precision and recall in the task's result. For the task of trust relationship prediction, or more generally friend recommendation, in an online social network that exists in an online social service, there are some considerations that might help with making the decision.

In OSNs, friend relationship or trust relationship recommendation is an efficient way for users to get started and engaged more in the online social service's activities so that the online social service's revenue could gain.

From the user's perspective, the recommendation should provide users with accurate sets of users based on their homophily and existing friend/trust networks, and the expected precision of the recommendation or the prediction is to be high ideally.

On the other hand, it is probable that the number of candidates being recommended to a user will be few due to the user's strict criteria, or nearly none, which would eventually make the recommendations less effective; and thus, from the online service's operating and managing perspective, in addition to suggested users who satisfy the user's criteria, it is essential to recommend extra possible candidates to the user, although these candidates may not strictly match the user's criteria fed to the inference algorithm. Therefore, a higher recall for such a recommender or a predictor of a model will benefit an online social network. Additionally, of course, the model should maintain acceptable precision.

Another reason for a higher recall comes from the fact that the dataset used for friend recommendation or trust relationship prediction tasks is usually incomplete. As previously discussed, part of an "ideally complete" dataset would be missing due to various actual reasons. Consequently, even though a model generates good prediction results with high precision, the results may still be far away from a "real ideal." Thus, as a trade-off, a prediction result with a higher recall is acceptable.

Taking the above considerations into account, one of the best models that fit in the trust relationship prediction tasks in this paper would achieve a higher recall while still maintaining an acceptable precision. From the results for the first experiment set, our proposed model had an average recall of 0.9327 in all 40 experiments; and for SVM, decision tree, and random forest in their respective 35 experiments, the average recalls were 0.8239, 0.7950, and 0.8382.

4.3.2. The Second Set of Experiments with Reduced Feature Data

From a quick look at the experiment setup, the second set of privacy-aware experiments conducted with reduced feature data revealed some similarities with the first set of experiments in the training set and test set split configuration. It is true that if we are to conduct another group of experiments in which users choose to opt out of usages of all their profile data and UGC data and even some of users deny the usage of their trust relationships in trust inference tasks, the experiments will be exactly the same as the ones in the first set of experiments using only the first category of auxiliary features ($\mathcal{F}_{\text{TAUX}}$) with specific training and test datasets.

However, the fact is that they are not the same from the management perspective. The split training set and test set configuration is for model verification. On one hand, the aforementioned experiments are only particular ones in trust relationship inference in this paper. On the other hand, real-world privacy settings vary a lot from service to service, OSN to OSN, and it is worth exploring how a model performs in close to real conditions.

Overall performance results

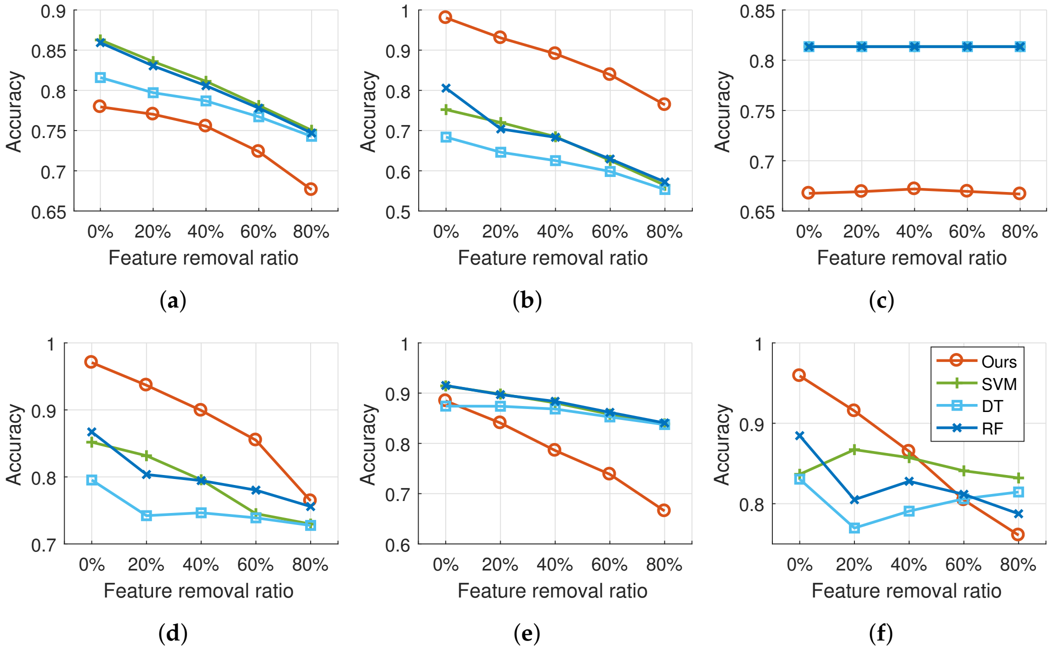
Figures 7 and 8 show the results of the second experiment set with reduced feature data evaluated by the Accuracy and the F_1 score metrics, respectively. In general, for different feature set combinations, the performance of the proposed model and the comparison methods decreases when the ratio of removed features increases.

Similarly to the results of experiments with full feature data, the proposed model works well with UGC features and outperformed other methods greatly in these experiments. This is the answer to the early question about the model's performance with partially available UGC data.

A second peek at Recall with reduced feature data

This set of experiments were a simulation of the real-world scenario where obtainable online social network datasets are incomplete due to various reasons. In particular, one whole category or certain types of data from a certain number of users are unavailable, when the users choose to limit the usage of their data by online social services through privacy settings. Although such an experimental setup in this paper does not cover all possible scenarios, it may shed some light on the study about how a model might work when some of the privacy-restricted data are unavailable.

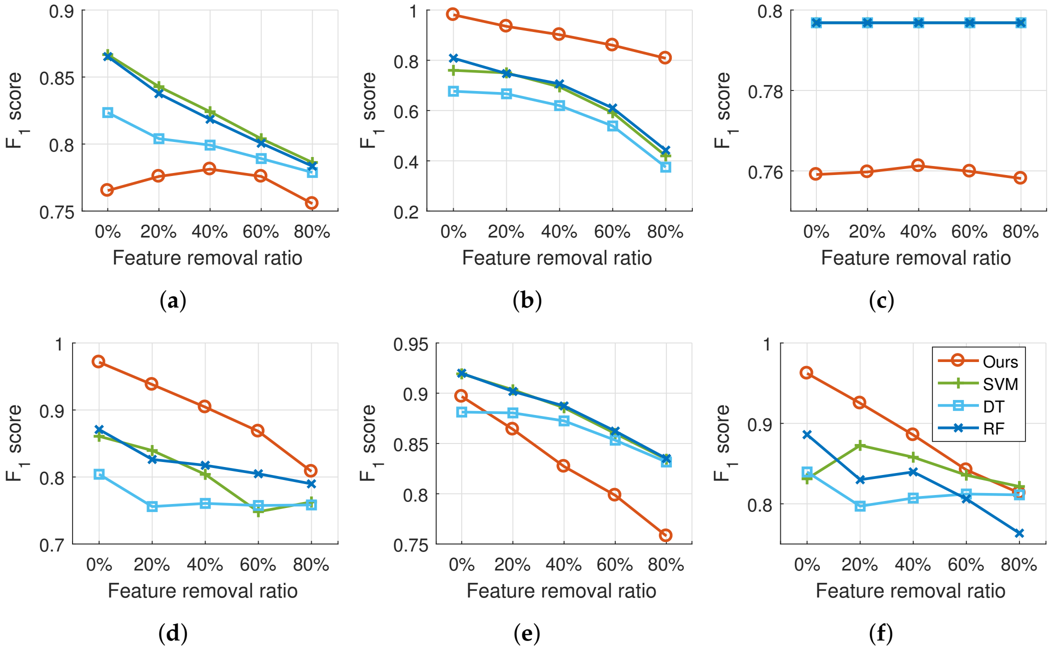
As previously discussed, a higher recall with acceptable precision achieved by a method to infer trust in OSNs is one of the ideal objects for online social services. In all 30 experiments from the second set, our proposed model achieved an average recall of 0.9409; 27 in 30 recalls were $\geq$0.90. The results for SVM, Decision Tree and random forest were, respectively: 0.7682 (2 recalls were ≥0.90), 0.7444 (none was ≥0.90), and 0.7844 (two recalls were ≥0.90). Refer to Supplementary Materials for the full results.



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

Figure 7. Performance results of the proposed model and comparison methods evaluated by accuracy for the second set of experiments. (**a**) Experiment 2s (Acc). (**b**) Experiment 3s (Acc). (**c**) Experiment 4s (Acc). (**d**) Experiment 5s (Acc). (**e**) Experiment 6s (Acc). (**f**) Experiment 7s (Acc).



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

Figure 8. Performance results of the proposed model and comparison methods evaluated by F_1 score for the second set of experiments. (**a**) Experiment 2s (F_1). (**b**) Experiment 3s (F_1). (**c**) Experiment 4s (F_1). (**d**) Experiment 5s (F_1). (**e**) Experiment 6s (F_1). (**f**) Experiment 7s (F_1).

5. Conclusions

In this paper, we explored the problem of collecting trust information and exploiting trust's various properties to infer trust in online social networks. Both explicitly presented trust relationships and information inexplicitly embedded in user-generated contents that bear users' attitude, experience, expertise, credibility, trustworthiness, etc., are harvested as trust information. A probabilistic graphical model based on a conditional random field for trust inference was proposed, which can effectively take advantage of trust's asymmetric, propagative, non-transitive, and subjective properties. With loopy belief propagation, a message passing algorithm, the model inference was presented and well interpreted. Experiments were conducted to evaluate the proposed model on a real-world online social trust dataset, and the experimental results demonstrated the effectiveness of the proposed model for trust inference.

Further improvements to the proposed model can be achieved. Implementing the model inference on cliques rather than pairwise structures may help the model to capture interplay among trust relationships that have same trustors or same trustees more accurately, making it possible to integrate more types of features that convey users' beliefs through complex interactions between users. Our handling of class imbalance for classifications, the underlying fact of which is the sparsity in user relationships in online social networks, is quite simple, and it may be further addressed by introducing penalties.

The study presented in this paper is primitive, but the proposed model is promising. Distrust is also a trust relationship type that can be supported by adding an extra label to the trustRelation node in the proposed model. Then, with a proper dataset, a model that can infer relationships of both trust and distrust can be trained. The proposed model also supports quantitative and context-specific trust evaluation, which could be an interesting future study with a proper dataset. By using each user's probability of trusting another user, an individual-oriented personalized trust management system can be built, and many social recommendation tasks will benefit from it.

With respect to trust's subjectivity and asymmetric properties, the concept of distinguishing the trustor's attitude, experience and belief from the trustee's expertise, trustworthiness, and credibility when forming a trust relationship—which was shown to be helpful for supervised trust inference in this paper—may help to improve unsupervised methods for trust inference. Finally, it will also be crucial to study the model's sensitivity against attacks.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/app12105186/s1. Detailed feature statistics used in this paper and full experimental results, including performances evaluated by accuracy, precision, recall and F_1 score for each experiment conducted in this paper, are enclosed in Supplementary Materials.

Author Contributions: Conceptualization, Y.L. and B.W.; data curation, Y.L.; methodology, Y.L.; software, Y.L.; validation, Y.L.; writing, Y.L.; funding acquisition, B.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partly funded by the National Key Research and Development Program of China under grant 2018YFC0831500, the National Natural Science Foundation of China under grant 61972047, and the NSFC-General Technology Basic Research Joint Funds under grant U1936220.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The full data of the results used to support the findings of this study are enclosed in the Supplementary Materials. The source code for our model implementation and raw data for the experiments conducted in this paper are available from the corresponding author upon request.

Acknowledgments: Yu Liu would like to thank Sergey Kosov for the inspiration for CRF network implementations and inference algorithms. The authors also would like to thank Yunlei Zhang and Jinna Lv for their kind and supportive suggestions for model implementation with accelerated computing using GPUs, and Bin Wu for providing a free test environment for GPU computing.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this paper:

OSN	Online Social Network
UGC	User-Generated Contents
CRF	Conditional Random Field
r.v.	random variable
TP	Trust Propagation
POS	Parts-of-Speech
BP	Belief Propagation
LBP	Loopy Belief Propagation
GPU	Graphics Processing Unit
SVM	Support Vector Machine
RBF	Radial Basis Function
DT	Decision Tree
RF	Random Forest

References

1. Golbeck, J. Trust on the World Wide Web: A Survey. *Found. Trends® Web Sci.* **2008**, *1*, 131–197. [CrossRef]
2. Meng, X.; Zhang, G. TrueTrust: A feedback-based trust management model without filtering feedbacks in P2P networks. *Peer-Netw. Appl.* **2020**, *13*, 175–189. [CrossRef]
3. Qolomany, B.; Mohammed, I.; Al-Fuqaha, A.; Guizani, M.; Qadir, J. Trust-Based Cloud Machine Learning Model Selection for Industrial IoT and Smart City Services. *IEEE Internet Things J.* **2021**, *8*, 2943–2958. [CrossRef]
4. Zhao, J.; Wang, W.; Zhang, Z.; Sun, Q.; Huo, H.; Qu, L.; Zheng, S. TrustTF: A tensor factorization model using user trust and implicit feedback for context-aware recommender systems. *Knowl.-Based Syst.* **2020**, *209*, 106434. [CrossRef]
5. Sparrowe, R.T.; Liden, R.C. Two Routes to Influence: Integrating Leader-Member Exchange and Social Network Perspectives. *Adm. Sci. Q.* **2005**, *50*, 505–535. [CrossRef]
6. Sherchan, W.; Nepal, S.; Paris, C. A Survey of Trust in Social Networks. *ACM Comput. Surv.* **2013**, *45*, 33. [CrossRef]
7. Searle, J.R.; Willis, S. *The Construction of Social Reality*; Simon and Schuster: New York, NY, USA, 1995.
8. Jøsang, A.; Ismail, R.; Boyd, C. A survey of trust and reputation systems for online service provision. *Decis. Support Syst.* **2007**, *43*, 618–644. Emerging Issues in Collaborative Commerce. [CrossRef]
9. Gupta, P.; Goel, A.; Lin, J.; Sharma, A.; Wang, D.; Zadeh, R. WTF: The Who to Follow Service at Twitter. In Proceedings of the 22nd International Conference on World Wide Web (WWW '13), Rio de Janeiro, Brazil, 13–17 May 2013; Association for Computing Machinery: New York, NY, USA, 2013; pp. 505–514. [CrossRef]
10. Sharma, S.; Menard, P.; Mutchler, L.A. Who to trust? Applying trust to social commerce. *J. Comput. Inf. Syst.* **2019**, *59*, 32–42. [CrossRef]
11. Golzardi, E.; Sheikhahmadi, A.; Abdollahpouri, A. Detection of trust links on social networks using dynamic features. *Phys. A Stat. Mech. Its Appl.* **2019**, *527*, 121269. [CrossRef]
12. Bathla, G.; Aggarwal, H.; Rani, R. A graph-based model to improve social trust and influence for social recommendation. *J. Supercomput.* **2020**, *76*, 4057–4075. [CrossRef]
13. Wu, L.; Sun, P.; Fu, Y.; Hong, R.; Wang, X.; Wang, M. A Neural Influence Diffusion Model for Social Recommendation. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'19), Paris, France, 21–25 July 2019; Association for Computing Machinery: New York, NY, USA, 2019; pp. 235–244. [CrossRef]
14. Zuo, L.; Xiong, S.; Qi, X.; Wen, Z.; Tang, Y. Communication-Based Book Recommendation in Computational Social Systems. *Complexity* **2021**, *2021*, 6651493. [CrossRef]
15. Elbeltagi, I.; Agag, G. E-retailing ethics and its impact on customer satisfaction and repurchase intention: A cultural and commitment-trust theory perspective. *Internet Res. Electron. Netw. Appl. Policy* **2016**, *26*, 288–310. [CrossRef]
16. Vosoughi, S.; Roy, D.; Aral, S. The spread of true and false news online. *Science* **2018**, *359*, 1146–1151. [CrossRef] [PubMed]
17. Zhang, B.; Zhang, L.; Mu, C.; Zhao, Q.; Song, Q.; Hong, X. A most influential node group discovery method for influence maximization in social networks: A trust-based perspective. *Data Knowl. Eng.* **2019**, *121*, 71–87. [CrossRef]
18. Chui, M.; Manyika, J.; Bughin, J. The Social Economy: Unlocking Value and Productivity through Social Technologies. McKinsey Global Institute. 1 July 2012. Available online: https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/the-social-economy (accessed on 14 May 2022).
19. Wu, J.; Xiong, R.; Chiclana, F. Uninorm trust propagation and aggregation methods for group decision making in social network with four tuple information. *Knowl.-Based Syst.* **2016**, *96*, 29–39. [CrossRef]
20. jiao Du, Z.; yang Luo, H.; dong Lin, X.; min Yu, S. A trust-similarity analysis-based clustering method for large-scale group decision-making under a social network. *Inf. Fusion* **2020**, *63*, 13–29. [CrossRef]

21. Ghafari, S.M.; Beheshti, A.; Joshi, A.; Paris, C.; Mahmood, A.; Yakhchi, S.; Orgun, M.A. A Survey on Trust Prediction in Online Social Networks. *IEEE Access* **2020**, *8*, 144292–144309. [CrossRef]
22. General Data Protection Regulation (EU) 2016/679 (GDPR). Available online: https://en.wikipedia.org/wiki/General_Data_Protection_Regulation (accessed on 10 April 2022).
23. California Consumer Privacy Act (CCPA). Available online: https://en.wikipedia.org/wiki/California_Consumer_Privacy_Act (accessed on 10 April 2022).
24. Personal Information Protection Law of the People's Republic of China. Available online: https://en.wikipedia.org/wiki/Personal_Information_Protection_Law_of_the_People's_Republic_of_China (accessed on 10 April 2022).
25. Schall, D. Link prediction in directed social networks. *Soc. Netw. Anal. Min.* **2014**, *4*, 157. [CrossRef]
26. Barbieri, N.; Bonchi, F.; Manco, G. Who to Follow and Why: Link Prediction with Explanations. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '14), New York, NY, USA, 24–27 August 2014; ACM: New York, NY, USA, 2014; pp. 1266–1275. [CrossRef]
27. Mao, C.; Xu, C.; He, Q. A cost-effective algorithm for inferring the trust between two individuals in social networks. *Knowl.-Based Syst.* **2019**, *164*, 122–138. [CrossRef]
28. Oh, H.K.; Kim, J.W.; Kim, S.W.; Lee, K. A unified framework of trust prediction based on message passing. *Clust. Comput.* **2018**, *22*, 2049–2061. [CrossRef]
29. Massa, P.; Avesani, P. Controversial users demand local trust metrics: An experimental study on epinions.com community. *AAAI* **2005**, *1*, 121–126.
30. Golbeck, J.; Hendler, J.A. FilmTrust: Movie recommendations using trust in web-based social networks. *CCNC. Citeseer* **2006**, *2006*, 282–286.
31. Liu, G.; Yang, Q.; Wang, H.; Lin, X.; Wittie, M.P. Assessment of multi-hop interpersonal trust in social networks by Three-Valued Subjective Logic. In Proceedings of the IEEE INFOCOM 2014—IEEE Conference on Computer Communications, Toronto, ON, Canada, 27 April–2 May 2014; pp. 1698–1706. [CrossRef]
32. Liu, G.; Chen, Q.; Yang, Q.; Zhu, B.; Wang, H.; Wang, W. OpinionWalk: An efficient solution to massive trust assessment in online social networks. In Proceedings of the IEEE INFOCOM 2017—IEEE Conference on Computer Communications, Atlanta, GA, USA , 1–4 May 2017; pp. 1–9. [CrossRef]
33. Tang, J.; Gao, H.; Hu, X.; Liu, H. Exploiting Homophily Effect for Trust Prediction. In Proceedings of the Sixth ACM International Conference on Web Search and Data Mining (WSDM '13), Rome, Italy, 4–8 February 2013; Association for Computing Machinery: New York, NY, USA, 2013; pp. 53–62. [CrossRef]
34. Yao, Y.; Tong, H.; Yan, X.; Xu, F.; Lu, J. MATRI: A Multi-Aspect and Transitive Trust Inference Model. In Proceedings of the 22nd International Conference on World Wide Web (WWW '13), Rio de Janeiro, Brazil, 13–17 May 2013; Association for Computing Machinery: New York, NY, USA, 2013; pp. 1467–1476. [CrossRef]
35. Zheng, X.; Wang, Y.; Orgun, M.; Zhong, Y.; Liu, G. Trust Prediction with Propagation and Similarity Regularization. In Proceedings of the AAAI Conference on Artificial Intelligence 2014, Québec City, QC, Canada, 27–31 July 2014; Volume 28.
36. Liu, G.; Li, C.; Yang, Q. NeuralWalk: Trust Assessment in Online Social Networks with Neural Networks. In Proceedings of the IEEE INFOCOM 2019—IEEE Conference on Computer Communications, Paris, France, 29 April–2 May 2019; pp. 1999–2007. [CrossRef]
37. Cho, J.H.; Chan, K.; Adali, S. A Survey on Trust Modeling. *ACM Comput. Surv.* **2015**, *48*, 1–40. [CrossRef]
38. Wang, J.; Jing, X.; Yan, Z.; Fu, Y.; Pedrycz, W.; Yang, L.T. A Survey on Trust Evaluation Based on Machine Learning. *ACM Comput. Surv.* **2020**, *53*, 1–36. [CrossRef]
39. Mukherjee, S.; Weikum, G.; Danescu-Niculescu-Mizil, C. People on Drugs: Credibility of User Statements in Health Communities. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '14), New York, NY, USA, 24–27 August 2014; ACM: New York, NY, USA, 2014; pp. 65–74. [CrossRef]
40. Mukherjee, S.; Weikum, G. Leveraging Joint Interactions for Credibility Analysis in News Communities. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM '15), Melbourne, Australia, 18–23 October 2015; ACM: New York, NY, USA, 2015; pp. 353–362. [CrossRef]
41. Mao, Y.; Shen, H. Web of Credit: Adaptive Personalized Trust Network Inference From Online Rating Data. *IEEE Trans. Comput. Soc. Syst.* **2016**, *3*, 176–189. [CrossRef]
42. Liu, H.; Lim, E.P.; Lauw, H.W.; Le, M.T.; Sun, A.; Srivastava, J.; Kim, Y.A. *Predicting Trusts among Users of Online Communities: An Epinions Case Study*; Association for Computing Machinery: New York, NY, USA, 2008; EC '08, pp. 310–319. [CrossRef]
43. Liu, Y.; Wang, B.; Wu, B.; Shang, S.; Zhang, Y.; Shi, C. Characterizing super-spreading in microblog: An epidemic-based information propagation model. *Phys. A: Stat. Mech. Its Appl.* **2016**, *463*, 202–218. [CrossRef]
44. Recasens, M.; Danescu-Niculescu-Mizil, C.; Jurafsky, D. Linguistic models for analyzing and detecting biased language. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, 4–9 August 2013; Volume 1: Long Papers, pp. 1650–1659.
45. De Albornoz, J.C.; Plaza, L.; Gervás, P. SentiSense: An easily scalable concept-based affective lexicon for sentiment analysis. In Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12), Istanbul, Turkey, 23–25 May 2012; European Language Resources Association (ELRA): Istanbul, Turkey, 2012.

46. Friedli, S.; Velenik, Y. *Statistical Mechanics of Lattice Systems: A Concrete Mathematical Introduction*; Cambridge University Press: Cambridge, UK, 2017. [CrossRef]
47. Yedidia, J.S.; Freeman, W.T.; Weiss, Y., Understanding belief propagation and its generalizations. In *Exploring Artificial Intelligence in the New Millennium*; Gerhard, L., Bernhard, N., Eds.; Morgan Kaufmann Publishers Inc.: Burlington, MA, USA, 2003; pp. 239–269.
48. Liu, Y.; Li, J.; Zhang, Y.; Lv, J.; Wang, B. A High Performance Implementation of A Unified CRF Model for Trust Prediction. In Proceedings of the 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Exeter, UK, 28–30 June 2018; pp. 848–853. [CrossRef]
49. Bottou, L. Stochastic Gradient Descent Examples on Toy Problems. 2010. Available online: https://leon.bottou.org/projects/sgd (accessed on 16 September 2017).