ARIMA

EPI sciences

# Time and Content aware Implicit Social Influence Estimation to enhance Trust-based recommender systems

**Armel Jacques NZEKON NZEKO'O**[*1,2], **Hamza ADAMOU**[1,2], **Thomas MESSI NGUELE**[1,2,3], **Bleriot Pagnaul BETNDAM TCHAMBA**[1]

[1]University of Yaounde I, FS, Computer Science Department, P.O. Box 812, Yaounde, Cameroun
[2]Sorbonne University, IRD, UMI 209 UMMISCO, F-93143, Bondy, France
[3]University of Ebolowa, HITLC, Computer Engineering Department, Cameroon

*E-mail : armel.nzekon@facsciences-uy1.cm

*DOI : 10.46298/arima.13328*
*Submitted on April 2, 2024 - Published on March 19, 2025*
Volume : **42** - Year : **2025**
Special Issue : **CRI 2023 - 2024**
Editors : *Paulin Melatagia, René Ndoundam, Kamel Barkaoui, Blaise Omer Yenke*

**Abstract**

Nowadays, e-commerce, streaming and social networks platforms play an important role in our daily lives. However, the ever-increasing addition of items on these platforms (items on Amazon, videos on Netflix and YouTube, posts on Facebook and Instagram) makes it difficult for users to select items that interest them. The integration of recommender systems into these platforms aims to offer each user a small list of items that match their preferences. To improve the performance of these recommender systems, some work in the literature incorporate explicit or implicit trust between platform users through trust-based recommender systems. Indeed, many of these works are based on explicit trust, when each user designates those whom they trust in the platform. But this information is rare in most real-world platforms. Thus, other work propose to estimate the implicit trust that each user can grant to another. However, work that estimates implicit trust does not take into account the temporal dynamics of users' past following actions and even less the fact that a user can influence another on one category of item and not on another. In this paper, we propose time and content aware strategies to estimate social influence of one user on another. The resulting time and content aware implicit trust are integrated to trust-based recommender systems build on K-Nearest Neighbors (KNN) and Graph-based techniques. Experiments done for rating predictions with KNN and Top-N recommendations with Graph model show that time and content aware implicit trust make it possible to improve the performance of the KNN according to the RMSE metric by 7% and 10%, and the performance of the graph model according to the NDCG@10 metric by 59% and 08% respectively on the Ciao and Epinions datasets.

**Keywords**

Trust-based recommender systems; Social influence; Time and content aware implicit trust.

# I INTRODUCTION

Over the last three decades, we have witnessed a great growth in the importance of online platforms in our daily lives, this is the case of social networks like Facebook, Instagram, streaming platforms like Youtube and Netflix and e-commerce platforms like Amazon or Alibaba. However, the constant addition of large numbers of items to these platforms (posts on social networks, items on e-commerce platforms, videos and songs on streaming platforms) makes it very difficult for a user to choose an item that interests them. This can spoil their experience on the platform and lead them to abandon it, which can have negative consequences on the turnover generated by the platform. It is to avoid such situations that recommender systems are designed and integrated into these digital platforms [11, 15, 18, 23, 24, 28].

Indeed, recommender systems make it possible to offer each user a small number of items most likely to interest them in the very near future. But these techniques can still be improved, and to improve their performance, some work on trust-based recommender systems integrates explicit trust information which requires each user to designate those in whom they trust [6, 17, 27]. Unfortunately, such information is rare on most digital platforms.

In order to have trust-based recommender systems that are not dependent exclusively on explicit trust, other work propose to estimate, using various calculation strategies, the implicit trust that a user can grant to another, and integrate it to trust-based recommender systems [19, 20, 26]. One of the strategies for estimating the implicit trust that a user $u$ grants to another user $v$, is based on the analysis of the following actions of $v$ by $u$, which corresponds to the calculation of the social influence that $v$ exerts on $u$ [22, 25].

However, none of the current work on the estimation of the social influence that $v$ exerts on $u$ takes into account the temporal dynamics of followership actions (the influence of $v$ on $u$ varies over time), and even less the fact that the influence of one user on another can depend on the category of item ($v$ can influence $u$ in terms of fashion, but not in terms of food). Such aspects should be considered in the processes of estimating social influences between users in order to deduce good implicit trust values to be integrated into trust-based recommender systems.

In this paper, we propose ways to estimate time and content aware social influences between users using the history of these users' past actions on items. This allows us to deduce the values of time and content aware implicit trust between users, and to know their relevance, we evaluate their impact by integrating them into trust-based recommender systems built from User-based k-Nearest Neighbors [10] and on bipartite graph models [25].

The rest of the paper is structured as follows: in section II, we present a state of the art on trust-based recommender systems and the ways in which trust is considered. Then in section III, we first present the strategy for calculating time and content aware implicit trust and how to integrate them into trust-based recommender systems. The experiments and the results obtained are detailed in the section IV. Finally, the paper ends with a conclusion in section V.

# II BACKGROUND AND TRUST-BASED RECOMMENDER SYSTEMS

In this section, we present existing work on trust-based recommender systems that which incorporates information about the trust that one user places in another. We first present collaborating filtering approach of recommender systems in sub-section 2.1. Then we present its extension which are the recommender system based on explicit trust in sub-section 2.2, and recommender systems based on implicit trust in sub-section 2.3.

## 2.1 Collaborative filtering approach for recommender systems

The collaborative filtering approach assumes that *"users who have had the same preferences in the past will have the same preferences in the future"*. This is the most used and most studied approach in the literature, particularly through memory-based techniques such as k-nearest neighbors [1, 5] and graph-based recommender systems [7, 14, 25].

Memory-based recommendation techniques rely on two main steps: the first to determine similarities (correlations) between users (or items) and the second to compute predictions. In this category, the most used techniques are $K$-nearest neighbors (KNN) [1, 5], and recommendation graphs [7, 14, 25] that we present in the next two subsections.

### *2.1.1 K-Nearest Neighbors based recommender systems*

There are two variants of K-Nearest Neighbors (KNN): the first is user-based and the second is item-based [1, 5]. In the user-based KNN technique, the first step is to determine the $K$-nearest neighbors of the target user $u$ and the second step is to combine the preferences of the latter to infer the item to recommend. In contrast, in the item-based KNN technique, for each item $i$ that the target user $u$ has not yet selected, we determine the $K$-nearest neighbors of $i$ in the first step, then we combine the preferences of $u$ for these $K$ items in order to estimate the preference rate of $u$ for $i$. In the following we consider only user-based KNN because the integration of information on trust is more natural there.

**Determination of nearest neighbors.** Suppose we want to estimate the preference of the target user $u$ for an item $i$. In the user-based KNN, the goal of this first step is to have a restricted set of $K$ users similar to the target user $u$. To do this, you must first choose a similarity measure such as Pearson correlation coefficient [12] or the Cosine distance to compute similarity $sim(u, u')$ between two users, and then set either the limit size $K$ of the neighborhood, or a similarity threshold $s$ necessary to consider another user as a neighbor of $u$.

**Computation of recommendations.** To compute the preference of the target user $u$ for an item $i$ that he has not yet selected, we calculate the weighted average of the known preferences of the neighbors $V_u$ of $u$ for the item $i$ using the following equation 1. Neighbors $u'$ who are most similar to $u$ have a greater influence on the choice of items to recommend to him.

$$preference(u, i) = \frac{\sum_{u' \in V_u} preference(u', i) \times sim(u, u')}{\sum_{u' \in V_u} sim(u, u')} \tag{1}$$

### *2.1.2 Graph-based recommender systems*

Recommendation graphs are easily interpretable and provide a natural and intuitive framework for different types of applications. It is for these reasons that we chose a recommendation graph as the basic recommender system in our work and it is only this type of system that we present in the remainder of this section.

**Construction of the graph.** The recommendation graph which is constructed from the binary score matrix is the classic bipartite graph (BIP). In this graph, each user as well as each item is represented by a node. When a user $u$ shows positive interest on an item $i$, the node of $u$ is connected to the node of $i$ by a bidirectional edge $(u, i)$. This graph is common for computing recommendations, following the example of the work of Baluja et al. [7] on the recommendation of videos on Youtube and those of Yan et al. on tweet recommendation [14].

**Computation of recommendations.** In the case of the classic bipartite graph, the hypothesis considered to calculate recommendations is based on the proximity of the target user $u$ with the next items he will select. Thus, the objective is to recommend N items that $u$ has not yet selected and to which it is as close as possible in the graph. Following this principle, most algorithms for calculating recommendations are based on a random walk in which the source node is the one associated with $u$: the most used algorithm for this task is the personalized PageRank [3] .

$$PR = \alpha \cdot M \cdot PR + (1 - \alpha) \cdot d \tag{2}$$

Equation 2 presents the iterative version of personalized PageRank, Where $PR$ is PageRank vector that contains the importance of each node at the end of the random walk; $M$ is the transition matrix of the bipartite graph; $\alpha$ is the damping factor; and $d$ is the personalized vector indicating which nodes the random walker will jump to after a restart. In other words, $d$ allows to initialize the weight of source nodes. This process favors the recommendation of item that are close to source nodes. To recommend to a user $u$, $d(u) = 1$ and $d(v) = 0$ if $v \neq u$.

### 2.2 Recommender systems based on explicit trust

Pure collaborative filtering techniques rely either on explicit rating matrix of the ratings that users give to items, or on binary matrices constructed from the history of user actions on item. In both situations, several other types of data are ignored. One of the highest types of ignored information is the trust information that one user places in another, and yet knowing that $u$ trusts $v$, allows us to infer that $u$ will replicate the selection actions of the same items as $v$ [6, 17, 27].

Explicit trust occurs when trust information between users is available and provided by the users themselves. For example, a user $u$ may openly declare that he trusts another user $v$. When such information is available, it is necessary to use it in recommender systems because users are more willing to accept recommendations from trusted friends, as Sinha et al. show [2].

It is by following this principle that Mei et al. [22] proposed to incorporate trust data to reinforce recommendations from trusted users while limiting the impact of others. They used trust data from Epinions to improve the K-nearest neighbor (KNN) model. They computed an inter-user trust matrix, then predicted user scores on items based on trusted neighbors. The formula for predicting user $u$'s score on item $i$ is given by the equation 3:

$$\hat{r}_{ui} = \mu_u + \frac{\sum_{v \in P_u(i)} Trust(u, v).(r_{vi} - \mu_v)}{\sum_{v \in P_u(i)} Trust(u, v)} \tag{3}$$

where $Trust(u, v)$ represents the trust that $u$ places in $v$, $P_u(i)$ corresponds to the set of users whom $u$ trusts and who have rated item $i$, and $\mu_u$ is the average of the marks given by $u$.

### 2.3 Recommender systems based on implicit trust

The work in the previous sub-section 2.2 clearly shows that the integration of explicit trust information between users makes it possible to improve the quality of recommender systems. However, this explicit information is rarely available on digital platforms. It is to overcome this limit that several work propose to estimate the weight of trust relationships between users for whom this is not explicitly defined: the resulting trust weights are the implicit trust [19, 20, 26].

In this section, we focus exclusively on the estimation of implicit trust because this is what marks the difference between recommendation systems based on explicit trust and those based

on implicit trust. Two scenarios are distinguished: the case where some explicit trust information is available, but in addition to this, trust propagation algorithms are used to infer implicit trust information. In the second case, explicit trust information is not available, and the objective is to use other information to construct a network with trust values between users.

### *2.3.1 Implicit trust deduced from explicit trust*

In a real world user network, laziness, lack of time and the fact that a user may have direct opinions about only a small number of users. To address this, techniques are developed to predict the reliability of unevaluated users based on existing trust information. These methods aim to predict trust scores between all pairs of network nodes, even those for which no explicit opinion has been provided. These approaches typically rely on the assumption of transitivity within the network: if $u$ trusts $v$, and $v$ trusts $w$, then $u$ trusts $w$.

In some work, trust information are binary, while in others, they are continuous. In a binary trust network, edges are categorized as *"approved"* or *"not approved"*. To infer the binary trust values $t_{sd}$ from the source $s$ to the target node $d$ a voting algorithm is proposed [16]. When the trust information are continuous values, the source queries its neighbors to obtain the trust weights assigned to the target node, repeating this process for each neighbor. When a node receives weights from multiple neighbors, it takes the average. This approach often forms the basis of others estimations of implicit trust. For example, when two users are connected via other users in a path, a propagation technique is used to calculate their mutual trust [8]. In [21], authors define alternative techniques for estimating trust between users.

### *2.3.2 Estimating implicit trust without using explicit trust*

Ziegler et al. [4] show that there is a relationship between similar user preferences and trust between them. This means that people who share the same interests and tastes tend to trust each other more. We can therefore conclude that it is reasonable to use measures of user preference similarity to infer implicit trust values. Most of these techniques are based on the similarity of user profiles (hey are linked in a social network, have friends in common) and the history of explicit ratings (users who assign similar ratings are more likely to trust each other).

Using the criterion of explicit rating history, Nzekon et al. [25] estimated the implicit trust between users of the Epinons and Ciao datasets and they integrated it in graph-based recommender systems. To estimate implicit trust, they used the Jaccard similarity $trust(u, v) = |I_u \cap I_v|/|I_u \cup I_v|$, where $I_u$ is the set of items purchased by user $u$. To recommend to user $u$, the personalized vector $d$ is configured as follows: $d(u) = 1-\gamma$, $d(v) = (\gamma \cdot trust(u, v))/|TR_u|$ if $v \in TR_u$ and $d(v) = 0$ otherwise; $\gamma$ is the parameter that allows to calibrate the influence of trusted users, and $TR_u$ is the set of users trusted by $u$.

This way of estimating trust between users does not require explicit trust information. Instead, it relies on the history of users' past actions on items. This is reassuring because this information is very frequently available on digital platforms. However, this approach to estimating implicit trust has some limitations. First, it appears to treat trust as a symmetrical relationship, when in reality trust is an asymmetrical relationship. For example, if a user $u$ trusts $v$, that does not mean that $v$ trusts him. Furthermore, these existing measures of implicit trust do not take into account the fact that the trust placed in someone varies over time and even sometimes depends on certain categories of items ($u$ can be influenced by $v$ in fashion, but not in food).

## III TIME AND CONTENT AWARE IMPLICIT SOCIAL INFLUENCE ESTIMATION

In this section, we first propose time and content aware strategies to estimate implicit social influence of one user on another. The resulting time and content aware implicit trust are integrated to trust-based recommender systems build on K-Nearest Neighbors and Graph-based models. The figure 1 shows the general architecture of our work. In the left side, we choose the social influence matrix: *Hub* for asymmetric social influence, *Seq* social influence that considers the order in the sequence of user actions, and the last three (*TimeE, TimeL, TimeP*) integrate time decay functions. It is also possible to consider the fact that a user can influence another on one item category and not on another by integrating (*Cat*) in the process. Then integrate the resulting implicit trust into the recommender system *KNN* or *Graph* models on the right side.

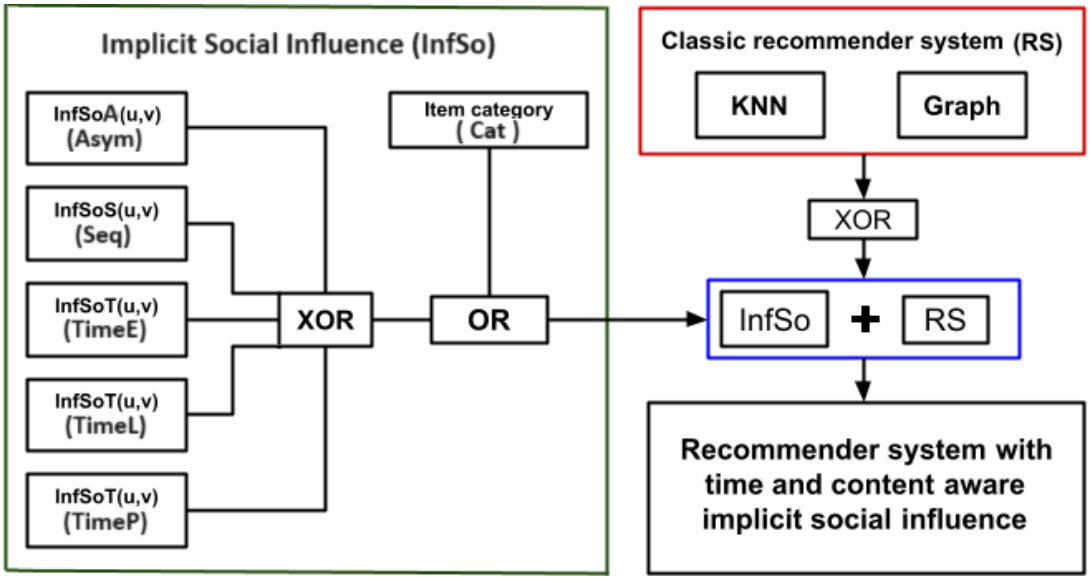


Figure 1: Architecture of the proposed trust-based recommender system integrating implicit trust information resulting from time and content aware implicit social influence estimation.

### 3.1 Estimation of time and content aware implicit social influence

This sub-section presents asymmetric implicit social influence, time aware implicit social influence and the time and content aware implicit social influence estimations.

#### *3.1.1 Asymmetric implicit social influence estimation*

Nzekon et al. [25] propose the use of Jaccard measure to estimate the trust between two users $u$ and $v$. But the fact that this measure is symmetric means that it does not reflect reality, because the trust that $u$ places in $v$ is not always the same that $v$ places in $u$. It is for this reason that we propose to modify the Jaccard equation to express the asymmetric nature of trust. Equation 4 is the result of the modification made, and $I_u$ is the set of items that user $u$ has rated positively.

$$InfSoA(u,v) = \frac{|I_u \cap I_v|}{|I_u|} \tag{4}$$

Figure 2 (a) illustrates the idea of asymmetric implicit trust estimation. Here, for example, we can see that both users enjoyed six items in common, but that $u$ enjoyed almost all of his items in common with $v$, while $v$ enjoyed only a tiny fraction of his items in common with $u$. We can therefore say that $u$ is more influenced by $v$ than $v$ is influenced by $u$.

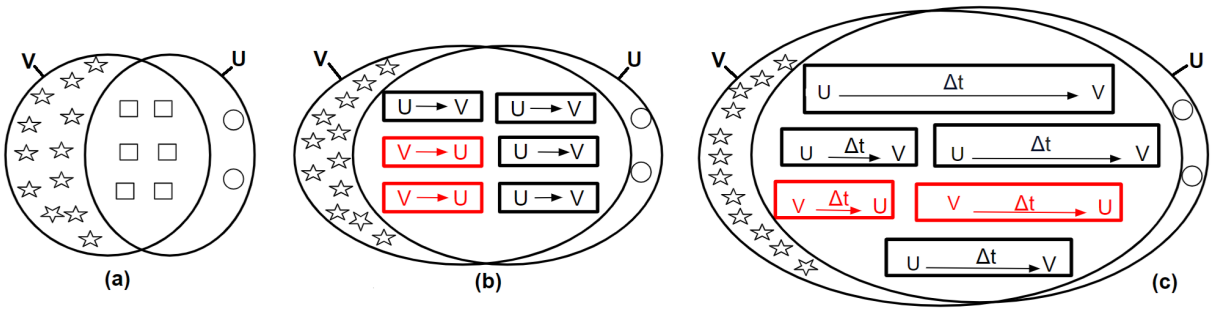


Figure 2: Different levels of consideration of following actions between users $u$ and $v$: (a) Asymmetric social influence, (b) Social influence with sequences order, (c) Time aware implicit social influence.

### *3.1.2 Sequence aware implicit social influence estimation*

The asymmetric estimation of implicit social influence as described above does not take into account the order of operations carried out by $u$ and $v$ for the items they have in common. Indeed, if we estimate the influence of $v$ on $u$, it is not relevant to count the items that $u$ purchased before $v$. To address this limit, we adjust the formula, incorporating the condition that $u$ must have purchased the item after $v$. The equation 5 is the result of the modification, and $|u \rightarrow v|$ is the length of the set of items purchased or selected by $u$ after $v$.

$$InfSoS(u,v) = \frac{|u \rightarrow v|}{|I_u|} \tag{5}$$

Figure 2 (b) illustrates the idea of sequence aware implicit social influence estimation. Among the six items selected by $u$ and $v$, it's notable that for four of them, $u$ selected items after $v$, while for the other two items marked in red, the reverse is true. So, in applying this new version, we only consider the four items for which $u$ made the purchase after $v$.

### *3.1.3 Time aware implicit social influence estimation*

Consider the case where $u$ buys the item two hours after $v$, and also consider the case where $u$ buys the item two years after $v$. Comparing the two cases, we can affirm that $v$ exerts a greater influence on $u$ in the first case and less in the second case, because the faster $u$ replicates the action of $v$, more we can say that $u$ is under the influence of $v$. However, the previous equations do not take this detail into account. We propose through the equation 6, an estimation of implicit social influence which integrates the time elapsed before the following action of $u$.

$$InfSoT(u,v) = \frac{\sum_{(u,v)\in\{u\rightarrow v\}} f(t_u - t_v)}{|I_u|} \tag{6}$$

$\{u \rightarrow v\}$ is the set of cases where $v$ selected the item before $u$, $t_v$ and $t_u$ are respectively the times at which $v$ and $u$ selected the item. The function $f()$ is a temporal decay function whose role is to reduce the weight of following actions as their associated duration increases.

In this work, we consider three decay functions : Exponential decay function (*TimeE*) $f(x) = e^{-x \cdot ln(2)/T_0}$, Logistic decay function (*TimeL*) $f(x) = 1 - 1/(e^{-K(x-T_0)} + 1)$ and Power decay function (*TimeP*) $f(x) = x^{(log_{T_0}(1/2))}$. $T_0$ is the duration after which a weight decreases by half.

Figure 2 (c) illustrates Time aware implicit social influence estimation. $\Delta t$ refers to the duration between the moment when $v$ selected the item and the moment when $u$ replicated the selected action of $v$. These durations are computed only when $u$ selected the item after $v$.

### 3.1.4 *Time and content aware implicit social influence estimation*

The influence that a user $v$ exerts on a user $u$ can vary depending on the type of items concerned. Indeed, $v$ can exert an influence on $u$ in fashion, and not have an influence on $u$ in the choice of his foods. To take this aspect into account, the influences between users are computed by item category. Therefore, all user actions on items are partitioned to have as many subsets as item categories. Then the different strategies presented previously are applied to each subset.

Equations 7, 8, and 9 present the calculations made for time and content aware social influence.

$$InfSoA\_Cat(u,v,c) = \frac{|I_{uc} \cap I_{vc}|}{|I_{uc}|} \tag{7}$$

$$InfSoS\_Cat(u,v,c) = \frac{|u \to v|_c}{|I_{uc}|} \tag{8}$$

$$InfSoT\_Cat(u,v,c) = \frac{\sum_{(u,v)\in\{u\to v\}_c} f(t_u - t_v)}{|I_{uc}|} \tag{9}$$

Where $c$ is the concerned item category, $I_{uc}$ is the set of items selected by $u$ and which belong to category $c$, $\{u \to v\}_c$ is the set of items that $u$ selected after $v$ and that belong to the $c$ category, $|u \to v|_c$ is the length of $\{u \to v\}_c$ and $t_u$ is the time at which $u$ selected the item.

## 3.2 Integration of implicit social influence into classic recommender systems

In this section, we show how to integrate the social influences in K-Nearest Neighbors User-based collaborative filtering and in classic Bipartite recommender graph.

### 3.2.1 *K-nearest neighbors - KNN*

To integrate implicit trust into KNN user-based collaborative filtering, it is only the prediction function that is modified as suggested by Mei et al. [22] through the equation 3. In this paper we did the same, and therefore for each type of implicit social influence estimated, we simply replaced $Trust()$ with the desired social influence as presented in equation 10. Where $InfSo()$ can be any implicit social influence compute using equations 4, 5 and 6.

$$\hat{r}_{ui} = \mu_u + \frac{\sum_{v\in P_u(i)} InfSo(u,v).(r_{vi} - \mu_v)}{\sum_{v\in P_u(i)} InfSo(u,v)} \tag{10}$$

Concerning the case of taking into account content information as item categories, the prediction formula is modified to consider only the users who influence the target user on the categories of the concerned item. The result is given by the equation 11. $InfSo\_Cat(u,v,c))()$ can be any time and content aware implicit social influence compute using equations 7, 8 and 9.

$$\hat{r}_{ui} = \mu_u + \frac{\sum_{c\in Cat(i)} \sum_{v\in P_u(i)} InfSo\_Cat(u,v,c).(r_{vi} - \mu_v)}{\sum_{c\in Cat(i)} \sum_{v\in P_u(i)} InfSo\_Cat(u,v,c)} \tag{11}$$

$Cat(i)$ is the set of categories of item $i$. For the choice of neighborhood, when the item belongs to several categories, we use the average of the influences of all the categories of item $i$.

### 3.2.2 *Bipartite recommender graph*

To integrate the time aware implicit trust into the classic bipartite recommender graph, it is only the process of computing recommendations with the personalized PageRank which is modified following the approach of Nzekon et al. [25] as presented in equation 2. The main idea is to adjust the personalized vector $d$ and we did it as presented below:

$$d(u) = 1 - \gamma \text{ and } d(v) = \frac{\gamma \cdot infSo(u,v)}{\sum_{v \in Q_u} infSo(u,v)} \text{ if } v \in Q_u \text{ and } 0 \text{ otherwise}$$

$InfSo()$ can be any implicit social influence compute using equations 4, 5 and 6. The symbol $u$ is the target user, $Q_u$ is the set of nodes associated to users who influence $u$, and $\gamma$ is the parameter that calibrates the weight of the influence of all $v \in Q_u$ on the target user $u$.

Concerning the case of taking into account item categories, the constructed graph is modified by integrating a new type of node which represents the association of a user with a item category $(u, c)$. These new nodes are connected to the items of category $c$ that user $u$ has selected. Figure 3 presents an example of such graph with three users, two items and two item categories.

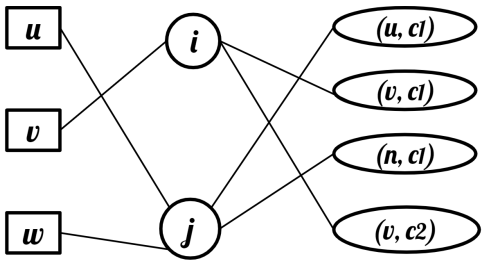


Figure 3: Bipartite graph with new user-content nodes $(u, c)$ related to item of category $c$ that user $u$ has selected. Item $i$ has two categories $\{c_1, c_2\}$ and item $j$ has one category $\{c_1\}$.

The calculation of predictions is done as before with the personalized PageRank with the only difference that the new nodes $(u, c)$ of the target user, and the new nodes $(v, c)$ of those who influence $u$, are also considered when personalizing the PageRank as presented below:

$$d(u) = 1-\gamma \text{ and } d(v) = \frac{\gamma \cdot infSo\_Cat(u,v,c)}{\sum_{c \in Cat} \sum_{v \in Q_u \cup Q_{uc}} infSo\_Cat(u,v,c)} \text{ if } v \in Q_u \cup Q_{uc} \text{ and } 0 \text{ otherwise}$$

$InfSo\_Cat(u, v, c))()$ can be any time and content aware implicit social influence compute using equations 7, 8 and 9. $Cat$ is the set of all item categories, and $Q_{uc}$ is the set of new user-content nodes of target user $u$ and all other users $v$ who influence $u$.

## IV EXPERIMENTS AND RESULTS

In this section we describe experiments and results obtened in the process of evaluating the impact of new ways of estimating implicit social influence for trust-based recommender systems.

### 4.1 Considered Dataset

We consider two datasets respectively from Ciao and Epinions [13] which are widely used in state-of-the-art on trust-based recommendation systems. These datasets contain the explicit trust network, which allows us to compare our results to those obtained with explicit trust. The Table 1 provides information on these two datasets. $min_u$ and $min_i$ denotes the minimum number of appearances of each user and each item. $|U|$, $|I|$ and $|C|$ refer, respectively, to the number of users, items and item categories present in the dataset. While $|R_{u,i}|$ and $|Trust_{u,v}|$ quantifies respectively the number of ratings and explicit trust information present in the dataset.

| | $\|U\|$ | $\|I\|$ | $min_u$ | $min_i$ | $\|R_{u,i}\|$ | $Period$ | $\|C\|$ | $\|Trust_{u,v}\|$ |
|---|---|---|---|---|---|---|---|---|
| **Ciao** | 889 | 9053 | 1 | 1 | 12742 | 2007-2011 | 6 | 23385 |
| **Epinions** | 728 | 18141 | 20 | 2 | 58717 | 2006-2010 | 24 | 1381 |

Table 1: Description of the Epinions and Ciao datasets.

### 4.2 Evaluation protocol

To evaluate the impact of our proposals, we use time aware k-fold cross validation as in [9, 25]. Suppose that the initial instant of the dataset is $t_0$ and that the duration which covers the initial training set is $\Delta_{train}$ and the fixed duration of a test set is $\Delta_{test}$. The training set of rank $k$ is defined in the interval $[t_0; t_0 + \Delta_{train} + (k-1) \times \Delta_{test}]$ and the test set of rank $k$ is in the interval $[t_0 + \Delta_{train} + (k-1) \times \Delta_{test}; t_0 + \Delta_{train} + k \times \Delta_{test}]$. In the experiments carried out in this work, we fixed $\Delta_{train}$ to 03 years, $\Delta_{test}$ to 04 months and perform 3 iterations.

For each couple training and test set of rank $k$, we compute the numerator $M_{num_k}$ and the denominator $M_{deno_k}$ of each considered evaluation metric $M_k$. Once we have all the numerators and denominators, we combine them into the Time Averaged (TA) value of the concerned evaluation metric: $TA(M) = \frac{\sum_k M_k \times M_{deno_k}}{\sum_k M_{deno_k}}$.

#### *4.2.1 Evaluation metrics*

We considered the two main tasks of recommender systems: rating prediction with the user-based KNN collaborative filtering and Top-N recommendation with classic Bipartite recommender graph. The metrics considered for the rating prediction are MAE and RMSE, and those used for the Top-N recommendations are Hit-Ratio and NDCG. The lower the rating prediction metrics, the better the recommendation system, and on the contrary, the higher the Top-N recommendation metrics, the better the recommendation system.

**MAE - Mean Absolute Error.** Refers to the mean of prediction errors absolute values. The prediction error here is the difference between the explicit user rating and the predicted rating. $MAE = \frac{\sum_{u,i \in r_{test}} |r_{ui} - \hat{r}_{ui}|}{|r_{test}|}$ where $|r_{test}|$ is the number of predictable ratings in the test set.

**RMSE - Root Mean Square Error.** Corresponds to the root square of the average rating prediction errors. $RMSE = \sqrt{\frac{\sum_{u,i \in r_{test}} (r_{ui} - \hat{r}_{ui})^2}{|r_{test}|}}$.

**HR - Hit Ratio.** The *Hit ratio* is the proportion of users to whom the recommender system has made at least one good recommendation in the Top-N list. This metric is used to determine the proportion of satisfied users on the platform. It is given by $HitRatio@N = \frac{\sum_{u \in U} (hit_N(u) > 0)}{|U|}$.

**NDCG - Normalized Discounted Cumulative Gain.** Takes into account the order of the good recommendation in the top-N list of recommendations. *NDCG* is calculated by dividing the cumulative gain *(DCG)* of the list of recommended items by the *DCG* of the ideal recommendation list. This ideal recommendation list is the list where relevant items are ranked in the most optimal order, i.e. in the top positions. $NDCG@N(u) = \frac{DCG@N(u)}{IDCG@N(u)}$ With $IDCG@N(u)$ being the ideal *DCG* considering a recommendation of N items to user $u$. For all users of the dataset, the *NDCG* is given by $NDCG@N = \frac{\sum_{u \in U} NDCG@N(u)}{|U|}$.

*Discounted Cumulative Gain (DCG)* metric is defined by $DCG@N(u) = \sum_{i=1}^{N} \frac{Gain(i)}{\log_2(i+1)}$ where $Gain(i) = 1$ if the item at position $i$ is a good recommendation and 0 otherwise.

#### *4.2.2 Parameters of considered recommender systems*

To increase the chances of having the best performances of different trust-based recommender systems, we vary the values of the parameters linked to the basic models (KNN and Graph) and to the different strategies for estimating implicit social influence and its integration into recommender systems. Table 2 presents the ranges of considered values of each parameter.

| Symbol | Parameter description | Predefined values |
|---|---|---|
| $K$ | Number of neighbors of target user in KNN | 2, 3, 5, 10, 20, 30 |
| $\alpha$ | PageRank damping factor of PageRank | 0.25, 0.5, 0.75 |
| $T_0$ | Half-life of time decay functions | 30, 60, 120, 240, 360 days |
| $\gamma$ | Influence factor attributed to $u$ influencers | 0.1, 0.3, 0.5, 0.7, 0.9 |

Table 2: Predefined parameter values of considered trust-based recommender systems.

### 4.3 Results and comments

This section presents results performances of all trust-based recommender systems that integrate time and content aware implicit trust estimated by our proposed strategies. We also study the impact of each proposed implicit social influence estimation on recommender systems and end the section by a comparison with some state-of-the-art trust-based recommender systems.

#### *4.3.1 Results performances of all proposed trust-based recommender systems*

Table 3 shows all the results obtained in the Ciao and Epinions datasets for the basic KNN and Graph recommender systems without any consideration of trust information, and the results of all the implicit trust-based recommender systems estimated by our proposed strategies. In column, we have the basic model and the metric according to which the recommender system is evaluated, and in line we have the type of implicit trust integrated into the recommender system. In each cell of the table, we have the performance according to the metric in the column, and more a cell has a color tends towards white, better is the performance of the recommender system. On the other hand, the closer this color is to red, worse is the performance.

| CIAO | KNN | | Graph | |
|---|---|---|---|---|
| | MAE | RMSE | HR@10 | NDCG@10 |
| - | 0.73831 | 1.02394 | 0.04082 | 0.01296 |
| Asym | 0.71888 | 0.99333 | 0.04082 | 0.01504 |
| Seq | 0.70136 | 0.96306 | 0.04082 | 0.01316 |
| TimeE | 0.69944 | 0.96092 | 0.04422 | 0.01562 |
| TimeL | 0.68744 | 0.95384 | 0.05102 | 0.01729 |
| TimeP | 0.70093 | 0.96287 | 0.04422 | 0.01313 |
| Asym-Cat | 0.70691 | 0.97866 | 0.05442 | 0.02006 |
| Seq-Cat | 0.69686 | 0.95991 | 0.05102 | 0.01991 |
| TimeE-Cat | 0.69531 | 0.958 | 0.04762 | 0.02059 |
| TimeL-Cat | 0.6874 | 0.95377 | 0.05442 | 0.01979 |
| TimeP-Cat | 0.6967 | 0.95964 | 0.05102 | 0.01969 |

| EPINIONS | KNN | | Graph | |
|---|---|---|---|---|
| | MAE | RMSE | HR@10 | NDCG@10 |
| - | 0.90281 | 1.17958 | 0.04444 | 0.0149 |
| Asym | 0.84827 | 1.10318 | 0.04444 | 0.01499 |
| Seq | 0.84764 | 1.09948 | 0.04444 | 0.01403 |
| TimeE | 0.84797 | 1.10017 | 0.04444 | 0.01419 |
| TimeL | 0.83337 | 1.07178 | 0.04233 | 0.01378 |
| TimeP | 0.84767 | 1.09948 | 0.04444 | 0.01407 |
| Asym-Cat | 0.84769 | 1.09585 | 0.0455 | 0.01609 |
| Seq-Cat | 0.8479 | 1.09272 | 0.04021 | 0.01431 |
| TimeE-Cat | 0.84816 | 1.09322 | 0.04127 | 0.01323 |
| TimeL-Cat | 0.8296 | 1.06226 | 0.04127 | 0.01397 |
| TimeP-Cat | 0.84791 | 1.09269 | 0.04233 | 0.01424 |

Table 3: Results performances of all proposed trust-based recommender systems in Ciao and Epinions

### *4.3.2 Best performances of proposed trust-based recommender systems*

Table 4 presents the comparison in Ciao and Epinions datasets, between the results of KNN and Gaph basic recommender systems and the best results obtained by integrating implicit trust deduced from proposed strategies of time and content aware implicit social influence estimation. **Imp.** represents the percentage of improvement or decrease of the performance that we have compared to the performance of basic KNN and Graph recommender systems.

| | CIAO - KNN | | EPINIONS - KNN | | CIAO - Graph | | EPINIONS - Graph | |
|---|---|---|---|---|---|---|---|---|
| | MAE | RMSE | MAE | RMSE | HR@10 | NDCG@10 | HR@10 | NDCG@10 |
| **BASIC** | 0.73831 | 1.02394 | 0.90281 | 1.17958 | 0.04082 | 0.01296 | 0.04444 | 0.0149 |
| **BEST** | **0.6874** | **0.95377** | **0.8296** | **1.06226** | **0.05442** | **0.02059** | **0.0455** | **0.01609** |
| **imp. (%)** | 07% | 07% | 08% | 10% | 33% | 59% | 02% | 08% |

Table 4: Percentage improvement of our best results compared to those of basic recommender systems.

We notice that our proposals always make it possible to improve the performance of basic recommendation systems. For example, we improve the performance of the KNN according to the RMSE metric by 7% and 10%, and the performance of the graph model according to the NDCG@10 metric by 59% and 08% respectively on the Ciao and Epinions datasets. This shows the relevance of taking the temporal dynamic of following user actions and item categories into account in the process of estimating implicit social influence between users.

### *4.3.3 Impact of each proposed implicit social influence to achieve the best performance*

In this subsection, we compare each proposed implicit social influence and evaluate the number of times it is among the best according to the four metrics and the two datasets (08 cases).

**Asym - Asymmetric implicit social influence.** Its results are better than those of the basic recommendation system in 6/8 (75%), but it is never the best in 08 cases.

**Seq - Sequence aware implicit social influence.** Its results are better than those of the basic recommendation system in 5/8 (63%), but it is never the best in 08 cases.

**Time - Time aware implicit social influence.** Its results are better than those of the basic recommendation system in 18/24 (75%), but it is never the best in 08 cases.

**Asym_Cat - Asymmetric and content aware implicit social influence.** Its results are always better than those of the basic recommendation system, and it is the best in 3/8 (37.5%).

**Seq_Cat - Sequence and content aware implicit social influence.** Its results are better than those of the basic recommendation system in 6/8 (75%), but it is never the best in 08 cases.

**Time_Cat - Time and content aware implicit social influence.** Its results are better than those of the basic recommendation system in 18/24 (75%), and it is the best in 6/8 (75%).

From these observations, we can conclude that to be sure to do better than the the basic recommendation system, you must choose the $Asym\_Cat$ option, but to maximize your chances of having the best performance, it is better to use $Time\_Cat$ and particularly $TimeL\_Cat$ (Logistic decay function) according to the results obtained on the two datasets considered.

#### *4.3.4 Comparison with some state-of-the-art trust-based recommender systems*

In this subsection, we compare the best results obtained with the proposed time and content aware implicit social influence, to some state-of-the-art results on trust-based recommender systems. We consider the work of Nzekon et al. [25] which propose to use the Jaccard similarity measure to estimate implicit trust (*IT Jaccard*), and integrates explicit trust in a binary way as provided in the Ciao and Epinions datasets (*ET Bin*). We also consider the work of Mei et al. [22] which propose two ways to integrate explicit trust into basic recommender systems, one which considers the activity level of those whom the target user $u$ trusts (*ET Activ*), and the other which considers the degree of trust that other users place in the one $u$ trusts (*ET Deg*).

| | CIAO - KNN | | EPINIONS - KNN | | CIAO - Graph | | EPINIONS - Graph | |
|---|---|---|---|---|---|---|---|---|
| | MAE | RMSE | MAE | RMSE | HR@10 | NDCG@10 | HR@10 | NDCG@10 |
| **-** | 0.73831 | 1.02394 | 0.90281 | 1.17958 | 0.04082 | 0.01296 | 0.04444 | 0.0149 |
| **IT Jaccard** | 0.71973 | 0.99342 | 0.84592 | 1.09995 | 0.04082 | 0.01364 | 0.04233 | 0.01448 |
| **ET Bin** | 0.719 | 0.99745 | **0.82376** | 1.04357 | 0.02381 | 0.00753 | 0.04233 | 0.0138 |
| **ET Activ** | 0.72047 | 0.99811 | 0.82378 | 1.04352 | 0.04422 | 0.01741 | 0.04127 | 0.01353 |
| **ET Deg** | 0.71952 | 0.99762 | 0.82377 | **1.04351** | 0.01701 | 0.00682 | 0.04021 | 0.0131 |
| **Our Result** | **0.6874** | **0.95377** | 0.8296 | 1.06226 | **0.05442** | **0.02059** | **0.0455** | **0.01609** |

Table 5: comparison of our best results with those of some state-of-the-art strategies.

Table 5 presents the comparison of our best results with those of some state-of-the-art strategies *IT Jaccard*, *ET Bin*, *ET Activ* and *ET Deg* for the two datasets Ciao and Epinions and for the four evaluation metrics considered, MAE, RMSE, HR@10 and NDCG@10. Best performances are in bold font. And we notice that out of all 08 possible cases, we have the best performance in 6/8 (75%) of possible cases. And more precisely in 100% of cases in the Ciao dataset and in 50% of cases in the Epionions dataset.

When we look at the percentage of improvement compared to the best performance of the state-of-the-art, we see that the-state-of-the-art is better than us by -0.7% and -2% in the cases of the KNN model in Epinions, respectively following the MAE and RMSE metrics. The improvements we make are 4%, 4%, 23% and 18% in the Ciao dataset respectively following the MAE, RMSE, HR@10 and NDCG@10 metrics. In Epinions, we do better than the best of the state of the art by 2% and 11% following the HR@10 and NDCG@10 metrics for the graph model.

## V CONCLUSION

In this paper, we aimed to improve trust-based recommender systems. To do this, we propose to integrate the temporal dynamics of users' following actions and the influence based on the item categories in the process of estimating implicit social influence. We carried out experiments on two reference datasets Ciao and Epinions, considering two memory-based recommender systems (KNN and Graph) and four recommendation evaluation metrics for rating prediction (MAE, RMSE) and Top-recommendations (HitRatio@10, NDCG@10).

The results obtained show for example that, we improve the performance of the KNN according to the RMSE metric by 7% and 10%, and the performance of the graph model according to the NDCG@10 metric by 59% and 08% respectively on the Ciao and Epinions datasets. This shows the relevance of the time and content aware implicit social influence that we proposed. To further evaluate the relevance of our work, we compared ourselves to some state-of-the-art strategies that integrate both explicit and implicit trust, and we found that the recommendation systems based on confidence that we propose are better than the existing ones in 75% of cases.

By comparing the different approaches proposed in our work, we see that From these observations, we can conclude that to be sure to do better than the basic recommendation system, you must choose the $Asym\_Cat$ option, but to maximize your chances of having the best performance, it is better to use $Time\_Cat$ and particularly $TimeL\_Cat$ (Logistic decay function) according to the results obtained on the two datasets considered.

As a perspective of this work, we plan to use implicit social influence propagation algorithms exploiting the transitivity of social influence to obtain a much denser implicit social influence matrix. In addition, we plan to integrate social influences into model-based recommender systems such as Matrix Factorization, Support Vector Machines and Artificial Neural Networks. Another perspective of this work would be to conduct experiments on other datasets.

## REFERENCES


[1] G. Karypis. "Evaluation of item-based top-n recommendation algorithms". In: *Proceedings of the tenth international conference on Information and knowledge management*. ACM. 2001, pages 247–254.

[2] R. R. Sinha, K. Swearingen, et al. "Comparing recommendations made by online systems and friends." In: *DELOS* 106 (2001).

[3] T. H. Haveliwala. "Topic-sensitive pagerank". In: *Proceedings of the 11th international conference on World Wide Web*. ACM. 2002, pages 517–526.

[4] C.-N. Ziegler and J. Golbeck. "Investigating correlations of trust and interest similarity-do birds of a feather really flock together". In: *Decision Support Systems* 42.3 (2005), pages 1111–1136.

[5] J. Wang, A. P. De Vries, and M. J. Reinders. "Unifying user-based and item-based collaborative filtering approaches by similarity fusion". In: *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM. 2006, pages 501–508.

[6] Y. Wang and J. Vassileva. "A review on trust and reputation for web service selection". In: *27th international Conference on distributed computing systems workshops (ICDCSW'07)*. IEEE. 2007, pages 25–25.

[7] S. Baluja, R. Seth, D. Sivakumar, Y. Jing, J. Yagnik, S. Kumar, D. Ravichandran, and M. Aly. "Video suggestion and discovery for youtube: taking random walks through the view graph". In: *Proceedings of the 17th international conference on World Wide Web*. 2008, pages 895–904.

[8] F. E. Walter, S. Battiston, and F. Schweitzer. "A model of a trust-based recommendation system on a social network". In: *Autonomous Agents and Multi-Agent Systems* 16 (2008), pages 57–74.

[9] N. Lathia, S. Hailes, and L. Capra. "Temporal collaborative filtering with adaptive neighbourhoods". In: *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. 2009, pages 796–797.

[10] F. Zhang, L. Bai, and F. Gao. "A user trust-based collaborative filtering recommendation algorithm". In: *International Conference on Information and Communications Security*. Springer. 2009, pages 411–424.

[11] R. Zhou, S. Khemmarat, and L. Gao. "The impact of YouTube recommendation system on video views". In: *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. 2010, pages 404–410.

[12] P. Sedgwick. "Pearson's correlation coefficient". In: *Bmj* 345 (2012).

[13] J. Tang, H. Gao, and H. Liu. "mTrust: Discerning multi-faceted trust in a connected world". In: *Proceedings of the fifth ACM international conference on Web search and data mining*. 2012, pages 93–102.

[14] R. Yan, M. Lapata, and X. Li. "Tweet recommendation with graph co-ranking". In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics. 2012, pages 516–525.

[15] B. Shapira, L. Rokach, and S. Freilikhman. "Facebook single and cross domain data for recommendation systems". In: *User Modeling and User-Adapted Interaction* 23 (2013), pages 211–247.

[16] M. A. Abbasi, J. Tang, and H. Liu. "Trust-aware recommender systems". In: *Machine Learning book on computational trust, Chapman & Hall/CRC Press* (2014).

[17] R. Forsati, M. Mahdavi, M. Shamsfard, and M. Sarwat. "Matrix factorization with explicit trust and distrust side information for improved social recommendation". In: *ACM Transactions on Information Systems (TOIS)* 32.4 (2014), pages 1–38.

[18] C. A. Gomez-Uribe and N. Hunt. "The netflix recommender system: Algorithms, business value, and innovation". In: *ACM Transactions on Management Information Systems (TMIS)* 6.4 (2015), pages 1–19.

[19] S. Gupta and S. Nagpal. "Trust aware recommender systems: a survey on implicit trust generation techniques". In: *International Journal of Computer Science and Information Technologies* 6.4 (2015), pages 3594–3599.

[20] X. Ma, H. Lu, and Z. Gan. "Implicit trust and distrust prediction for recommender systems". In: *Web Information Systems Engineering–WISE 2015: 16th International Conference, Miami, FL, USA, November 1-3, 2015, Proceedings, Part I 16*. Springer. 2015, pages 185–199.

[21] A. Selmi, Z. Brahmi, and M. M. Gammoudi. "Trust-based recommender systems: an overview". In: *Proceedings of 27th international business information management association (IBIMA) conference, Milan, Italy*. 2016.

[22] J.-P. Mei, H. Yu, Z. Shen, and C. Miao. "A social influence based trust model for recommender systems". In: *Intelligent Data Analysis* 21.2 (2017), pages 263–277.

[23] B. Smith and G. Linden. “Two decades of recommender systems at Amazon. com”. In: *Ieee internet computing* 21.3 (2017), pages 12–18.

[24] Q. Chen, H. Zhao, W. Li, P. Huang, and W. Ou. “Behavior sequence transformer for e-commerce recommendation in alibaba”. In: *Proceedings of the 1st international workshop on deep learning practice for high-dimensional sparse data*. 2019, pages 1–4.

[25] A. J. N. Nzeko’o, M. Tchuente, and M. Latapy. “A general graph-based framework for top-N recommendation using content, temporal and trust information”. In: *Journal of Interdisciplinary Methodologies and Issues in Sciences* 5 (2019).

[26] A. Zahir, Y. Yuan, and K. Moniz. “AgreeRelTrust—a simple implicit trust inference model for memory-based collaborative filtering recommendation systems”. In: *Electronics* 8.4 (2019), page 427.

[27] S. Medjroud, N. Dennouni, and M. Loukam. “Recommendation System: a review of trust techniques”. In: *NCAIA’2023* (2023), page 84.

[28] Y.-J. Wang and A. J. Lee. “Movie account recommendation on Instagram”. In: *ACM Transactions on Internet Technology* 23.1 (2023), pages 1–21.

## A ACKNOWLEDGEMENTS

The successful completion of this research has been made possible through the generous support and collaboration of the International Development Research Centre (IDRC), Swedish International Development Cooperation Agency (SIDA) and African Center for Technology Studies (ACTS) for awarding the Artificial Intelligence for Development (AI4D) scholarship programme that funded this research. This scholarship not only provided financial support but also served as a source of motivation and encouragement throughout the project.

We are grateful for the support of all members of our work team namely HIgh PERformance DAta Science (HIPERDAS), who contributed to stimulating discussions and shared their insights. The collaborative environment fostered by our academic community has been integral to the development of this work. Lastly, we extend our thanks to all those who, directly or indirectly, played a role in the realization of this paper.