applied sciences

Article

A Model for Deriving Trust and Reputation on Blockchain-Based e-Payment System [†]

Jaehong Ahn [1], Mingyu Park [2], Hyungsik Shin [3] and Jeongyeup Paek [2,*]

[1] School of European Languages and Cultures, Chung-Ang University, Seoul 06974, Korea; ahong94@cau.ac.kr
[2] Department of Computer Science and Engineering, Chung-Ang University, Seoul 06974, Korea; hello0922@cau.ac.kr
[3] School of Electronic and Electrical Engineering, Hongik University, Seoul 04066, Korea; hyungsik.shin@hongik.ac.kr
[*] Correspondence: jpaek@cau.ac.kr

[†] An earlier and shorter preliminary version of this article was presented as a *poster* in the International Conference on ICT Convergence (ICTC) 2018, Jeju, Korea, 17–19 October 2018.

Received: 1 November 2019; Accepted: 3 December 2019; Published: 8 December 2019

Abstract: E-commerce has become a crucial part of our life allowing us to buy products, request services, and transfer money easily with a press of a button. As such, establishing immutable trust and reputation of entities that are resilient to manipulation by the malicious are critical in today's online systems. In this work, we propose a model for calculating trust and reputation using the values stored on blockchain ledger. The model is applied to blockchain-based online payment systems which have a characteristic of immutability by preventing data manipulation. The model normalizes user evaluations based on each user's personal evaluation criteria that changes over time. In addition, the model derives reputation of, and trust between, users by applying psychological factors. We evaluate our model using not only simulated transaction data but also on real Bitcoin transaction-based dataset to show that our model is able to derive stable values from immutable transactions on blockchain-based online payment systems. Our model has been built into a live commercial blockchain service platform, and new application developments are underway.

Keywords: e-Commerce; trust; reputation; blockchain; bitcoin

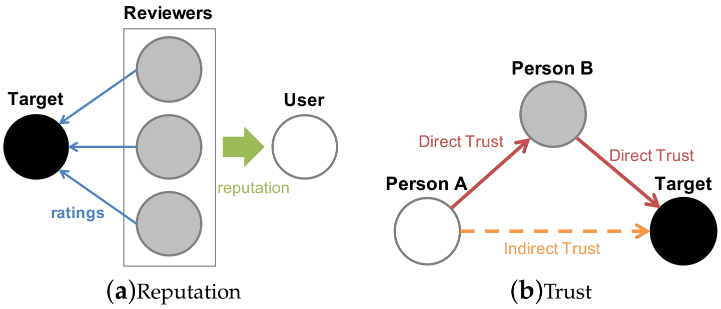
1. Introduction

Advancements in e-commerce technology over the past few years have enabled customers to easily purchase products or services, or even transfer money, at the end of their fingertips over the Internet. As a result, e-commerce market has expanded with a very rapid growth rate, which is reflected by the number of transactions taking place all around the world for a day. Usually when a customer shops for a product, he or she considers various factors such as price, quality, delivery time, etc. When different sellers offer a same product at similar prices or conditions, then the next thing that customers pay attention to are the seller's *reputation* or the degree of *trust* to those sellers. Many existing online payment systems often do provide such information, but information such as reputation are prone to manipulation by malicious entities, which include the advertisers or owners themselves who may give extremely high or low ratings on purpose. Furthermore, as each individual has different evaluation criteria, the value of reputation could be non-objective and unreliable. Therefore, it is difficult for customers to distinguish between good and bad products, or between reliable and unreliable sellers. Thus, as online markets start to play important roles in our daily life, building an online payment system that are trustworthy has become an important issue. The question is, how?

In 2008, Satoshi Nakamoto presented Bitcoin [1], a decentralized digital cryptocurrency based on blockchain technology. Blockchain keeps data in a list of blocks in which blocks are sequentially linked (pointing to the previous block), time-consuming to create, and almost impossible to overtake the competitors by several blocks. This chain is created in a distributed manner by numerous competing peers over the network. Thus the data stored in blocks eventually become invulnerable to potential attacks that might try to manipulate them. Due to these unique characteristics, and also in line with Bitcoin's recent success, there has been growing interest in applying blockchain technology to various domains, such as digital rights management [2–4], health care [5–8], Internet of things [9–13] and e-commerce [14,15]. In particular, applying blockchain's immutability to online payment systems can make it difficult for malicious entities to manipulate transaction history such as the amounts or ratings, which will result in better reliability, security, and trustworthiness of the systems.

To explore the design of a trustworthy online e-commerce system, we implemented a blockchain-based electronic payment system where not only the transaction data, but also the mutual and bi-directional "rating" information is recorded into the blockchain. For the ratings, the parties of the transactions evaluate each other in the range from 1 to 10, or equivalently from one to five stars with half star granularity, based on their satisfaction after every transaction. For instance, say a consumer Alice purchases a product or a service from a provider Bob. After the transaction is validated, Alice grades the quality of the product/service. Similarly, Bob, the seller, also evaluates Alice based on his own criteria. For example, if Alice frequently asks for a refund without any specific reason, Bob might give her low evaluation, which may mean defining her as a black consumer. Then, these ratings are linked to their corresponding transactions within the blockchain after every transaction gets validated, as if it is a transaction of its own in its virtual ratings currency. The goal is to build a reliable method for measuring trustworthy *reputation* and degree of *trust* of entities participating in e-commerce transactions based on their immutable history.

In general, the concepts of *reputation* and *trust* may look similar but they are fundamentally different in the way they work. Reputation is formed based on the word-of-mouth (WoM) mechanism [16]. As depicted in Figure 1a, evaluations of every person who has done transactions with the target can influence on target's reputation. On the other hand, trust is a subjective probability—a degree that affects an individual's decision and judgment—that is formed between a person and a target. Direct trust is derived if there had been transactions between the person and the target. Otherwise, indirect trust is derived as a second option via any other person who had relations with both parties. Figure 1b illustrates this where indirect trust from person A to the target is derived via person B. Indirect trust is not as influential as the direct trust. However, it could serve as a friends' suggestion which could help customers make a reasonable judgment about a target for the first time. In that sense, trust is often explained with a keyword friend-of-a-friend (FoaF) [17].



(**a**)Reputation (**b**)Trust

Figure 1. Reputation and trust model.

In this paper, we suggest a model for calculating *reputation* and *trust* values efficiently from a history of transactions and rating information. We maintain the history on blockchain-based payment system for its immutability, but for efficiency, our model does not parse all the previous transaction data per each transaction update that took place on the blockchain-based payment system. Instead, we utilize a small cache of key values to expedite the queries, while periodically validating and verifying those values in the background for reliability without deteriorating user experience. We evaluate our model using both the data from our online payment system (system is real, but transaction/rating data are created for testing) as well as synthetic rating data derived from real Bitcoin transaction history (rating is synthetic, but transaction is real and massive). The purpose of using Bitcoin transaction history is to validate the notion of time in calculation with large amount of data, and also to test the query–response and processing latency for parsing real blockchain data. With rating information given by users, our model can derive reliable reputation and trust values which could be useful for the users to make decisions before buying or selling products/services on the Internet. Also, those values could help both the merchants and customers prevent abnormal transactions.

The remainder of this paper is structured as follows. We first introduce related work in Section 2. Then, Section 3 discusses the design of our model and system for deriving reliable reputation and trust values from raw transactions and evaluation scores stored on blockchain-based system. In Sections 4 and 5, we show the efficiency of our model through experiments conducted using simulated dataset and actual Bitcoin transactions. Finally, Section 6 concludes the paper.

2. Related Work

There are several pieces of prior work that suggest the application of blockchain technology to e-commerce payment systems [14,15] or propose models for deriving reputation scores of the system users [18,19]. For example in 2015, Carboni presented a decentralized and distributed feedback management system built on top of the Bitcoin blockchain without changing the existing Bitcoin protocol [18]. This model, however, has a potential vulnerability of data manipulation, which could be attempted by malicious users giving extremely high or low scores on purpose. The author did suggest that the problem could be mitigated by charging the parties to pay (i.e., transfer money) to give feedback. However in a real world, who would pay just to give feedback? In 2016, Schaubs et al. suggested blockchain-based trustless reputation system which anonymizes the identity of users [19]. All evaluation data are safely stored in the blocks so that manipulation is nearly impossible, just like our system, and the system expects users to evaluate each other without worrying about retaliations. In real life, however, this approach may bring unwanted side effects. For example, in the Web communities operated on anonymity basis, users may slander others with no proper reason. As a result, the effect expected by using anonymity could be discouraged due to unreliable data generated on the system.

There are also trust models for e-commerce systems that are not based on blockchain [20,21]. E2CTM (Enhanced E-Commerce Trust Model) [20] suggested a trust model where a trustor uses personal experiences (e.g., purchase transactions) with a trustee and input from other trustors about that trustee to gain strust in trustees and come to a decision about whether to continue transactions with that trustee. Moreover, in the E2CTM model, if a trustee with a good reputation value abuses his good reputation and performs a fraudulent transaction, all other transactions with other trustors will be affected because it will be reported as soon as possible. PeerTrust [21] proposed a coherent adaptive trust model for quantifying and comparing the trustworthiness of peers based on a transaction-based feedback system, similar to ours. Suggested model introduces three basic parameters and two adaptive trust factors, the transaction context factor, and the community context factor, in computing trustworthiness of peers.

Building trust and reputation models suitable for the target application are not an easy task, and knowing the internal operations of a commercial system is also not possible in most cases. Resnick et al. [22] analyzed the eBay's repuation system, one of the earliest and best-known Internet reputation systems in e-Commerce, which gathers comments from buyers and sellers about each other after each transaction. The work reports several interesting features from examination of a large data set from 1999. However, there is no way to know the internal operations of eBay, and the work did not propose a new model. The work by Siau and Shen [23] argued that building customer trust is a complex process that involves technology and business practices, as well as movement from initial trust formation to continuous trust development.

As will be described later, our model can prevent the manipulation attacks by applying several techniques that carefully considers each user's grading tendency and dilute malicious ratings to transform the raw evaluation data into reliable and objective values. There have been prior works that suggest such standardization approach using various metrics. For example, use of each user's evaluation values' deviation [24], the min–max and *z*-score methods [25], and the *preference* model which takes account of each user's criteria for grading other users [26] has been proposed. Our model builds on top of these prior work, but goes a step further to consider the time, frequency, and human tendency in e-commerce transactions into account. Furthermore, we build our system on a blockchain-based system for immutability, with a periodic caching technique for low-latency query response and trustworthy validation process.

A separate line of research has been conducted on the stability or performance of the network nodes maintaining blockchains. To understand the performance of the Bitcoin network, Decker et al. experimentally observed that increasing network delays with PoW leads to increased forks in blockchain [27]. In 2014, Donet et al. reported a list of 872,648 IP addresses that run Bitcoin nodes, and studied the network stability and information propagation latency [28]. Later in 2018, Park et al. collected network data from Bitcoin network for 39 days, and analyzed geographical distribution, protocol/client version, and the type of nodes (full-node vs. lightweight-node) [29]. More recently, Seol et al. proposed a hierachical layered blockchain architecture and query-chaining mechanism to improve the performance (TPS) of blockchain while maintaining the distributed nature of the system and allowing for easier application development on top of blockchain [30].

There have been many proposals to develop services and systems based on the blockchain technology to take advantage of its properties such as immutability, irreversibility, and decentralization, and many more attempts are underway to utilize the technology [2–15,30,31]. For example, Xia et al. presented a platform to configure access control of medical information data through blockchain [7]. It utilizes blockchain's immutability to keep track of and record every access of individuals to patients' sensitive data. In addition, the structure for blocks and transactions were newly constructed to store medical information. Jeff Herbert et al. suggested a method for decentralized peer-to-peer software validation using cryptocurrency blockchain technology [32]. A user willing to get the license of a software sends a specific amount of tokens or money to the vendor on Bitcoin or Bespoke model. Both of these examples has similar goal and approach as ours; protect the valuable information (i.e., evaluation rating, medical, software information) via blockchain, and retrieve them efficiently.

3. Design

When dealing with thousands to millions of transaction evaluations for each block generation, there are three main issues to be considered. First, it requires a lot of memory space and computing power to look up previous transactions for each block generation when updating the current value. Thus, it is necessary to develop a mechanism that requires only a small footprint of past transactions in a local cache for quick updates. Second, in general, user behaviors and thoughts change over time. Therefore, it is necessary to keep track of the natural changes of every person's evaluation criteria over time by considering the time intervals and frequency of transactions. Finally, smoothing the gathered values via low-pass filtering is required to understand the general trend of each user's grading tendency rather than individual instances.

To overcome these challenges and derive reputation and trust values efficiently, our model uses two general mathematical concepts. One is the well-known exponentially weighted moving average (EWMA), but added with the notion of "time" to model the varying inter-transaction times and adapt to transactions' frequency. The other is the '1sigmoid" function, which allows us to normalize and set bounds on scattered values, and to transform them into values representing customers' propensity. After normalization, the model derives each individual's reputation and trust between individuals.

The data derived by the model are safely stored in each individual's private space which is also generated by the blockchain mechanism, and they are recalculated upon arrival of a newly generated transaction without reparsing all of the previous transactions. Further details follow.

3.1. Time Adaptive EWMA with Personal Evaluation Criteria

To model the personal transaction evaluation criteria that changes over time, we introduce four key metrics: time difference weight (TDW), personal evaluation criteria (PEC), normalized evaluation (NE), and average received normalized evaluation (ARNE).

Time Difference Weight (TDW): The concept of '1time" takes an important role in human life. Our thoughts and behaviors change over time as well. Thus, it is important to trace the changes of a person's personality (evaluation criteria in our case) and to weigh the latest value more while making sure that the past data is not overwhelmed by a small number of recent transactions. In order to trace the changes of each individual's evaluation tendency, we define a time-adaptive EWMA weighting parameter $\alpha(td(t))$ as,

$$\alpha\big(td(t)\big) = \begin{cases} 0.05, & td(t) \leq 7 \\ 0.1, & td(t) \leq 30 \\ 0.15, & td(t) \leq 180 \\ 0.2, & td(t) \leq 365 \\ 0.25, & td(t) > 365 \end{cases} \tag{1}$$

according to the time difference $td(t)$ between previous and current transaction evaluation times as illustrated in Figure 2. The threshold values for the evaluation time differences are one week ($\leq$7), one month ($\leq$30), six months ($\leq$180), one year ($\leq$365), and over a year ($>$365). As stated in the Equation (1), the longer the time-interval between transactions is, higher weight is applied on the latest transaction. This is to balance the influence of several recent transactions dominating the rating and aggregated long term statistics.

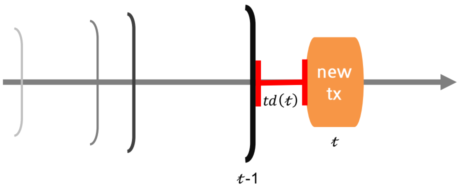


Figure 2. Time difference between transactions.

Personal Evaluation Criteria (PEC): Naturally, each individual has different evaluation tendency. For example, someone might be more generous or strict in evaluation, and some might have high or low variation in grades than others. To trace individual's evaluation tendency, we propose a personal evaluation criteria (PEC) metric based on the average and deviation of each user's evaluation scores. Our model updates these values for every new transaction, and store them in a local cache which can be verified at anytime by parsing the transaction history in the blockchain. By maintaining these values, the system can normalize evaluations in constant time, and it becomes unnecessary to look up all transactions of a user to find out her evaluation tendency.

$$PECavg_{A.tx}(t) = \begin{cases} e(t), & t = init \\ \alpha_{A.tx}\big(td(t)\big)\cdot e(t) + \Big(1 - \alpha_{A.tx}\big(td(t)\big)\Big)\cdot PECavg_{A.tx}(t-1), & t \neq init \end{cases} \tag{2}$$

$$PECdev_{A.tx}(t) = \begin{cases} 0, & t = init \\ \mid e(t) - PECavg_{A.tx}(t) \mid, & t = init + 1 \\ \alpha_{A.tx}\big(td(t)\big)\cdot \mid e(t) - PECavg_{A.tx}(t) \mid + \Big(1 - \alpha_{A.tx}\big(td(t)\big)\Big)\cdot PECdev_{A.tx}(t-1), & t > init + 1 \end{cases} \tag{3}$$

$$NE_{A.tx}(t) = \begin{cases} \frac{e(t)}{10}, & td(t) = init \\ 0.5, & PECdev_{A.tx}(t) = 0 \\ \frac{1}{1+e^{-\frac{(e(t)-PECavg_{A.tx})}{PECdev_{A.tx}}}}, & td(t) \neq init \end{cases} \tag{4}$$

$$ARNE_{B.rx}(t) = \begin{cases} NE_{A.tx}(t), & t = init \\ \alpha_{B.rx}\big(td(t)\big) \cdot NE_{A.tx}(t) + \Big(1 - \alpha_{B.rx}\big(td(t)\big)\Big) \cdot ARNE_{B.rx}(t-1), & t \neq init \end{cases} \tag{5}$$

Equation (2) is used to update the average of *PEC* upon every new transaction, where $e(t)$ is the latest "raw" evaluation value, *PECavg* is the long-term average up till now, and $\alpha_{A,tx}$ is the time-adaptive EWMA parameter for transactions made by user A. If the user evaluated a product/service for the first time, *PECavg* is initialized to $e(t)$. Otherwise, *PECavg* is calculated using EWMA with new evaluation $e(t)$ and $\alpha\big(td(t)\big)$ derived from the inter-grading times. Equation (3) calculates the latest deviation of *PEC* which represent the degree of variation on rating scores. *PECdev* is set to 0 if a user evaluates a transaction for the first time as there is no previous evaluation to compare with. Otherwise, it is computed using the same method as *PECavg* to keep track of changes on evaluation values.

Normalized Evaluation (NE): *NE* is an evaluation given by a user, normalized with sigmoid function based on her grading criteria *PEC* using Equation (4). After the normalization process, original rating values are converted into a range of 0 to 1. However, there are two exception cases to consider; If it is the first time that the user is giving a rating, *NE* is initialized with $e(t)/10$ to fit in the range of 0 to 1 as there is no previous transaction. When there are previously evaluated transactions but their values are all the same, *PECdev* will be zero, resulting in a situation in which normalization is not possible. In this case, a neutral value of 0.5 is used for *NE*, implying that "multiple ratings all with same value" is regarded as an indeterminable rating. After the normalization process, each *NE* value per transaction becomes the input for deriving *ARNE*, reputation of an individual, and trust of one to another.

Average Received Normalized Evaluation (ARNE): $ARNE_B$ is an average of all *NE* that user B has received from others, averaged using the time-adaptive EWMA parameter for transactions made by user B. As stated in Equation (5), $ARNE_B$ is initialized with $NE_{A,tx}(t)$ when user B is evaluated for the first time by user A. Thereafter, *ARNE* accumulates received *NE*s while the incoming value is weighted by time difference weight $\alpha\big(td(t)\big)$.

3.2. Reputation

$$Rep_{B.rx}(t) = \begin{cases} 0.5, & t = init \\ Rep_{B.rx}(t-1) + \alpha_{A\to B}\big(td(t)\big) \cdot Rep_{A.tx}(t) \cdot \big(NE_{A.tx}(t) - 0.5\big), & t \neq init \end{cases} \tag{6}$$

$$\beta_{A\to B}(t) = \begin{cases} 0.05, & NE_{A.tx}(t) \geq DTR_{A\to B}(t-1) \\ 0.25, & NE_{A.tx}(t) < DTR_{A\to B}(t-1) \end{cases} \tag{7}$$

$$DTR_{A\to B}(t) = \begin{cases} NULL, & \text{if there is no transaction} \\ 0.5, & t = init \text{ and not graded} \\ NE_{A.tx}(t), & t = init \text{ and graded} \\ \beta_{A\to B}(t) \cdot NE_{A.tx}(t) + \big(1 - \beta_{A\to B}(t)\big) \cdot DTR_{A\to B}(t-1), & t \neq init \end{cases} \tag{8}$$

$$ITR_{A\underset{C}{\to}B}(t) = \frac{DTR_{A\to C}(t) \cdot DTR_{C\to B}(t)}{DTR_{A\to C}(t) + DTR_{C\to B}(t)} \tag{9}$$

The main idea behind our reputation calculation is to not use the absolute rating values themselves (unlike most popular online shopping malls), but to only reflect the relative "positive" and "negative" opinion values of each review where the value is normalized on a per-reviewer basis. Specifically, every user's reputation is initialized to the *neutral value* 0.5 at the beginning. This value goes up and down according to the normalized evaluations she receives from others, offset by 0.5, as shown in Equation (6). Concurrently, $\alpha_{A \to B}(td(t))$ represents the *TDW* at the time when A evaluates B after the original transaction, and the new reputation of evaluatee B is weighted by the reputation of evaluator A. This is based on the intuition that *people tend to trust the evaluation of someone (as evaluator) who has higher reputation*. Last, *NE* is offset by 0.5 to use only the positive or negative "relative" grading aspect (per evaluator) with respect to *PEC* of A. For example, if *NE* is greater than 0.5, it means that A regards B better than others among those that A has evaluated. Of course, the final reputation value will be bounded to [0, 1].

3.3. Trust

In essence, trust can be thought of as a directional reputation of one by the other among two entities, where the relationship between the two can be either direct or indirect through another entity [20]. Only the entities (two for direct, three for indirect) involved in these relationship(s) will contribute to the calculation of trust. Other than that, the calculation will similarly be based on the *NE* and *TDW*. However, our model adds one extra concept to trust calculation: losing confidence is much easier than maintaining confidence. Generally in human life, we put a great deal of effort and time to get someone's trust. Same for businesses, and we apply this rule to the model.

Specifically, trust between user A and B are derived as follows. First, the model checks if there are any transactions between user A and B. If so, direct trust will be calculated. Otherwise, it calculates indirect trust between them via C who has done transactions with both A and B, if any (See Figure 1b). For every update of transactions, it checks again the existence of transactions between the two. If true, indirect trust is replaced by newly calculated direct trust.

Direct Trust (DTR): To derive *DTR*, the system first determines whether the evaluatee maintains the evaluator's trust or not as shown in Equation (7). $\beta_{A \to B}(t)$ represents the psychological factor of trust from A to B. If the normalized evaluation given by A is same or above the previous *DTR*, the weight β is set to 0.05 as it means the trust of A to B remains as before or improves. On the other hand, if lower, β is set to 0.25 as it could mean A loses confidence in B. If there is no transaction between A and B, *DTR* cannot be calculated (NULL) and *ITR* substitutes for its role. If there were transactions but A has never evaluated B, *DTR* is set to the neutral value 0.5. If the current evaluation is the first time that A grades B, *DTR* is initialized to *NE*. Otherwise, *DTR* is derived from the incoming *NE* values weighted by $\beta_{A \to B}(t)$.

Indirect Trust (ITR): When it comes to a situation where a customer has no *DTR* information (prior transaction) with a seller, the customer faces the following problem; "I do not know her. How can I trust her?" Although the customer can check the seller's reputation, it may not be enough for her to make a decision if the reputation is below her expectation. In this case, *ITR* shows off its ability. To calculate the *ITR* from A to B, our model determines a broker X who has the highest *DTR* product ($max(DTR_{A \to X} * DTR_{X \to B})$) among others who have done transactions with both A and B. Once the broker is determined, *ITR* is derived according to Equation (9). Figure 3 illustrates an example of *ITR* calculation between the "user" and the "target", given that there is no direct trust (past transaction) between the two. In this example, user A will be selected as the broker to calculate the *ITR*. If new *DTR* becomes available between the two entities, previously derived *ITR* is replaced by the newly generated *DTR*.

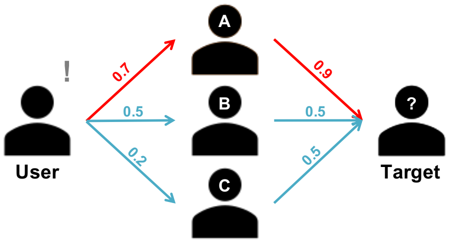


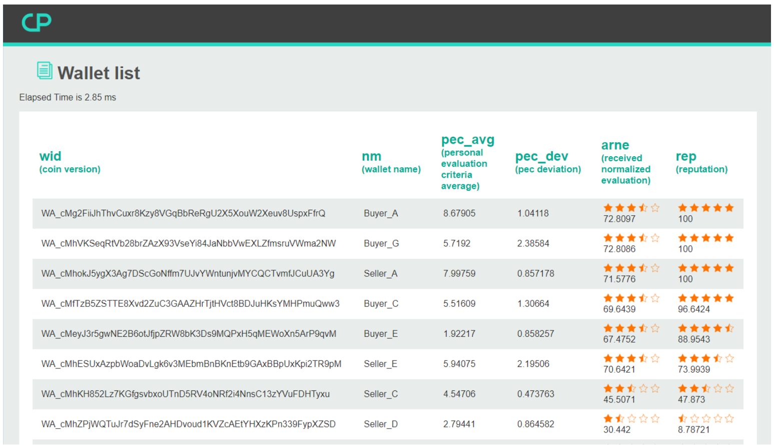
Figure 3. An example of indirect trust calculation between the "user" and the "target", given that there is no direct trust (past transaction and evaluation) between the two. In this example, user A will be selected as the broker to calculate the indirect trust.

3.4. Blockchain System and Caching

Our overall system is designed as follows. There is a blockchain-based electronic payment (escrow) system on which users can purchase products, make payments, and transfer money. In addition, the system allows the participating users to evaluate each other based on their satisfaction after every transaction. Then, not only the transaction data, but also the 'rating' information is recorded into the blockchain. These ratings are linked to their corresponding transactions within the blockchain after every transaction gets validated as if it is a transaction of its own in its virtual "ratings" currency.

Once we have the transaction and rating information in the blockchain, trust and reputation can be calculated simply by parsing those information in the blockchain and applying our model in Sections 3.1–3.3. For example, Figure 4 shows the web interface screenshot for the list of all wallets and their respective reputation/trust model values in our prototype escrow system based on blockchain. Furthermore, Figure 5a is the web screenshot for the detailed trust/reputation information of an user, and Figure 5b lists all transaction history of that user. We have built our model into a real blockchain-based prototype system for online transactions (including escrow service and money transfer), and this system is now evolving into a new blockchain-based platform (http://nodehome.io) that is preparing to launch soon as a commercial service.

Blockchain guarantees immutability, and thus the history is secure and reliable. However, parsing the whole blockchain for every transaction update would require significant amount of computation and time. For this reason, we propose to use a local cache that requires only a small footprint of key values for fast updates without the need for parsing all the previous transaction data per every transaction update that took place. However, there is a danger of local cache being out-of-sync with the original blockchain DB due to reasons such as system reboot, network disconnection, or even manipulation. To address this challenge, our system periodically validates and verifies the cache data in the *background* for reliability without deteriorating user experience.

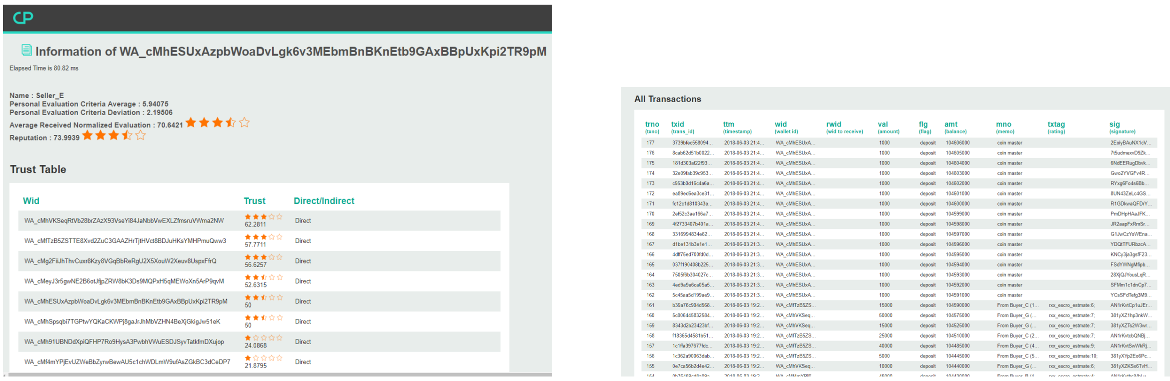


Wallet list

Elapsed Time is 2.85 ms

wid (coin version)	nm (wallet name)	pec_avg (personal evaluation criteria average)	pec_dev (pec deviation)	arne (received normalized evaluation)	rep (reputation)
WA_cMg2FiiJhThvCuxr8Kzy8VGqBbReRgU2X5XouW2Xeuv8UspxFfrQ	Buyer_A	8.67905	1.04118	★★★⯪☆ 72.8097	★★★★★ 100
WA_cMhVKSeqRtVb28brZAzX93VseYi84JaNbbVwEXLZfmsruVWma2NW	Buyer_G	5.7192	2.38584	★★★⯪☆ 72.8086	★★★★★ 100
WA_cMhokJ5ygX3Ag7DScGoNffm7UJvYWntunjvMYCQCTvmfJCuUA3Yg	Seller_A	7.99759	0.857178	★★★⯪☆ 71.5776	★★★★★ 100
WA_cMfTzB5ZSTTE8Xvd2ZuC3GAAZHrTjtHVct8BDJuHKsYMHPmuQww3	Buyer_C	5.51609	1.30664	★★★⯪☆ 69.6439	★★★★★ 96.6424
WA_cMeyJ3r5gwNE2B6otJfjpZRW8bK3Ds9MQPxH5qMEWoXn5ArP9qvM	Buyer_E	1.92217	0.858257	★★★⯪☆ 67.4752	★★★★⯪ 88.9543
WA_cMhESUxAzpbWoaDvLgk6v3MEbmBnBKnEtb9GAxBBpUxKpi2TR9pM	Seller_E	5.94075	2.19506	★★★⯪☆ 70.6421	★★★⯪☆ 73.9939
WA_cMhKH852Lz7KGfgsvbxoUTnD5RV4oNRf2i4NnsC13zYVuFDHTyxu	Seller_C	4.54706	0.473763	★★⯪☆☆ 45.5071	★★⯪☆☆ 47.873
WA_cMhZPjWQTuJr7dSyFne2AHDvoud1KVZcAEtYHXzKPn339FypXZSD	Seller_D	2.79441	0.864582	★⯪☆☆☆ 30.442	⯪☆☆☆☆ 8.78721

Figure 4. Web interface screenshot for listing all wallets in our blockchain system.



Information of WA_cMhESUxAzpbWoaDvLgk6v3MEbmBnBKnEtb9GAxBBpUxKpi2TR9pM

Elapsed Time is 80.82 ms

Name : Seller_E
Personal Evaluation Criteria Average : 5.94075
Personal Evaluation Criteria Deviation : 2.19506
Average Received Normalized Evaluation : 70.6421 ★★★⯪☆
Reputation : 73.9939 ★★★⯪☆

Trust Table

Wid	Trust	Direct/Indirect
WA_cMhVKSeqRtVb28brZAzX93VseYi84JaNbbVwEXLZfmsruVWma2NW	★★★☆☆ 62.2811	Direct
WA_cMfTzB5ZSTTE8Xvd2ZuC3GAAZHrTjtHVct8BDJuHKsYMHPmuQww3	★★★☆☆ 57.7711	Direct
WA_cMg2FiiJhThvCuxr8Kzy8VGqBbReRgU2X5XouW2Xeuv8UspxFfrQ	★★★☆☆ 56.6257	Direct
WA_cMeyJ3r5gwNE2B6otJfjpZRW8bK3Ds9MQPxH5qMEWoXn5ArP9qvM	★★⯪☆☆ 52.6315	Direct
WA_cMhESUxAzpbWoaDvLgk6v3MEbmBnBKnEtb9GAxBBpUxKpi2TR9pM	★★⯪☆☆ 50	Direct
WA_cMhSpsqbi7TGPtwYQKaCKWPj8gaJrJhMbVZHN4BeXjGkigJw51eK	★★⯪☆☆ 50	Direct
WA_cMh91UBNDdXpiQFHP7Ro9HysA3PwbhVWuESDJSyvTatkfmDXujop	★☆☆☆☆ 24.0868	Direct
WA_cMf4mYPjEvUZWeBbZyrwBewAU5c1chWDLmW9ufAsZGkBC3dCeDP7	★☆☆☆☆ 21.8795	Direct

All Transactions

trno (txno)	txid (trans_id)	ttm (timestamp)	wid (wallet id)	rwid (wid to receive)	val (amount)	flg (flag)	amt (balance)	mno (memo)	txtag (rating)	sig (signature)
177	3739bfec558094...	2018-06-03 21:4...	WA_cMhESUxA...		1000	deposit	104606000	coin master		2EoiyBAuNX1cV...
176	8cab62d51b0022...	2018-06-03 21:4...	WA_cMhESUxA...		1000	deposit	104605000	coin master		7t5udmexvD9Zk...
175	181d303af22f93...	2018-06-03 21:4...	WA_cMhESUxA...		1000	deposit	104604000	coin master		6NdEERugDbvk...
174	32e09fab39c953...	2018-06-03 21:4...	WA_cMhESUxA...		1000	deposit	104603000	coin master		Gwo2YVGFv4R...
173	c953b0d16c4a6a...	2018-06-03 21:4...	WA_cMhESUxA...		1000	deposit	104602000	coin master		RYxg6Fo4s6Bb...
172	ea89ed6ea3ce31...	2018-06-03 21:4...	WA_cMhESUxA...		1000	deposit	104601000	coin master		8UN43ZeLc4GS...
171	fc12c1d810343e...	2018-06-03 21:4...	WA_cMhESUxA...		1000	deposit	104600000	coin master		R1GDkwaQFDrY...
170	2ef52c3ae166a7...	2018-06-03 21:4...	WA_cMhESUxA...		1000	deposit	104599000	coin master		PmDHpHAaJFK...
169	4f2733407b401a...	2018-06-03 21:4...	WA_cMhESUxA...		1000	deposit	104598000	coin master		JR2aapFxRmSr...
168	3316994834e62...	2018-06-03 21:4...	WA_cMhESUxA...		1000	deposit	104597000	coin master		G1JwCzYoWEna...
167	d1be131b3e1e1...	2018-06-03 21:3...	WA_cMhESUxA...		1000	deposit	104596000	coin master		YDQtTFURbzcA...
166	4dff75ed700fd0d...	2018-06-03 21:3...	WA_cMhESUxA...		1000	deposit	104595000	coin master		KNCy3ja3gsfF23...
165	037f190408b225...	2018-06-03 21:3...	WA_cMhESUxA...		1000	deposit	104594000	coin master		FSdYWNgMfipb...
164	7505f6b304027c...	2018-06-03 21:3...	WA_cMhESUxA...		1000	deposit	104593000	coin master		28XjQJYousLqR...
163	4ed9a9e6ca05a5...	2018-06-03 21:3...	WA_cMhESUxA...		1000	deposit	104592000	coin master		SFMm1c1dnCp7...
162	5c45aa5d199ae9...	2018-06-03 21:3...	WA_cMhESUxA...		1000	deposit	104591000	coin master		YCsSFdTefg3M9...
161	b39a76c904d568...	2018-06-03 19:2...	WA_cMfTzB5ZS...		15000	deposit	104590000	From Buyer_C (1...	rxx_escro_estmate:6;	AN1rKvtCp1uJEr...
160	5c806445832584...	2018-06-03 19:2...	WA_cMhVKSeq...		50000	deposit	104575000	From Buyer_G (...	rxx_escro_estmate:7;	381yXZ1hp3nkW...
159	8343d2b23423bf...	2018-06-03 19:2...	WA_cMhVKSeq...		15000	deposit	104525000	From Buyer_G (...	rxx_escro_estmate:7;	381yXZTs2W3wr...
158	f18365d4581b51...	2018-06-03 19:2...	WA_cMfTzB5ZS...		25000	deposit	104510000	From Buyer_C (2...	rxx_escro_estmate:7;	AN1rKvtcbQNBj...
157	1c1ffa397677fdc...	2018-06-03 19:2...	WA_cMfTzB5ZS...		40000	deposit	104485000	From Buyer_C (4...	rxx_escro_estmate:9;	AN1rKvtSwWkRj...
156	1c362a90063dab...	2018-06-03 19:2...	WA_cMfTzB5ZS...		5000	deposit	104445000	From Buyer_C (5...	rxx_escro_estmate:10;	381yXYp2Eo6Pc...
155	0e7ca56b2d4e42...	2018-06-03 19:2...	WA_cMhVKSeq...		10000	deposit	104440000	From Buyer_G (...	rxx_escro_estmate:6;	381yXZKSx6TvH...
154	0b76469cd8a09a...	2018-06-03 19:2...	WA_cMf4mYPjE...		45000	deposit	104430000	From Buyer_B (4...	rxx_escro_estmate:4;	AN1rKvtbsiVbLu...

(**a**)Detailed trust/reputation info of a user. (**b**)List of all transactions of a user.

Figure 5. Web interface screenshots for a user in the blockchain-based payment & escrow system.

If any inconsistency is found between the cache and the blockchain data, new information derived from the blockchain DB overrides the cache. This is based on the fact that we are using blockchain; an immutable and irreversible distributed legder, given that concensus has been made and longest chain has been selected. In our prototype implementation, this periodic refresh and update occurs every 5 min, in addition to whenever the system admin triggers an manual update. However, this period is configurable based on the application requirement and load on the system. Figure 6 illustrates this design where the cache is periodically updated from the blockchain DB. The user can access the cache for fast query response instead of parsing the whole blockchain which may take significant time depending on the amount of data.

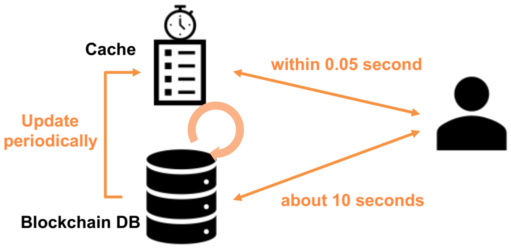
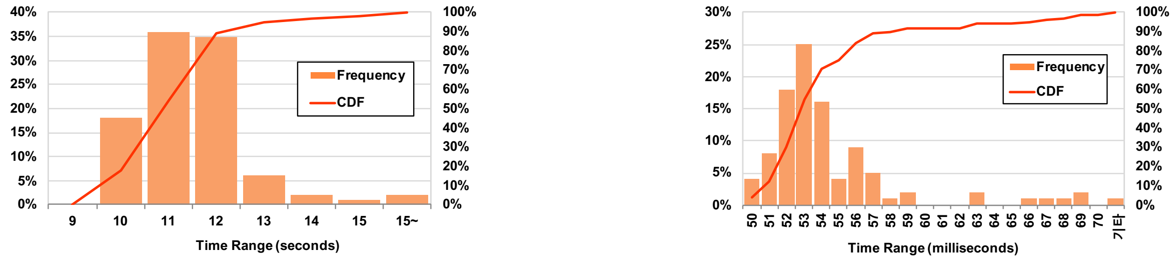


Figure 6. Illustration of the use of local cache for calculating trust and reputation from the blockchain DB.

To give an idea of the impact of using cache, we measured the actual transaction update latency (which includes trust and reputation calculation) on our system with the dataset used in Section 4. Figure 7 plots the distribution of thos latency measurements with (Figure 7b) and without (Figure 7a) the local cache. The two figures may look similar, but note the units of x-axis; they are three orders of magnitude different (seconds vs. milliseconds). This shows how critical it is to use the cache when looking up data in the blockchain DB.



(**a**)Without cache (from Blockchain DB)

(**b**)With cache

Figure 7. Distribution of latency measurements for calculating trust and reputation with and without the local cache for blockchain DB. Note that the units of x-axis is three orders of magnitude different (seconds vs. milliseconds).

4. Evaluation via Simulation with Controlled Dataset

To verify the effectiveness of the proposed model for deriving reputation and trust values, we first conduct controlled experiments with a dataset from our payment system consisting of various types of users. In the next section, we also evaluate the model using data from the actual Bitcoin transaction history. We compare our reputation and trust values with the naive "*cumulative average of raw evaluation*", which is used by most, if not all, e-commerce systems (e.g., average rating values on amazon.com, hotels.com, etc.). We expect that naive cumulative average of raw evaluation will not reflect the time factor and human tendency well. Note that the dataset generated on our local experimental setup are safely stored in a private blockchain-based storage built for the online payment system after applying our model.

Table 1 lists the evaluation criteria of each simulated buyer who grades sellers in the given range. In the table, Buyers A and B represent generous users who always give 7 to 10 as evaluation scores. On the other hand, Buyers C and D are strict in grading, giving only 1 to 4. Buyers E, F, G, and H stand for general users who evaluate transactions with wider ranges with a mean close to 5. Lastly, Buyers I and J grade others from 1 to 2 or 9 to 10, which represent extreme cases such as owners, advertisers, or those with decisive characteristics that could possibly exist in our real life. Furthermore, Table 2 shows five different simulated sellers that represent various types of providers. Specifically, the "good to bad" and "bad to good" sellers are expected to simulate their reputation changes over time according to decreasing and increasing evaluations they receive from counterparts, respectively. We generated twenty transactions for every buyer–seller pair, resulting in total of 1000 transactions. That is, each buyer and seller has evaluated hundred times and two hundred times, respectively, resulting in two-thousand transaction evaluations (ratings) in total.

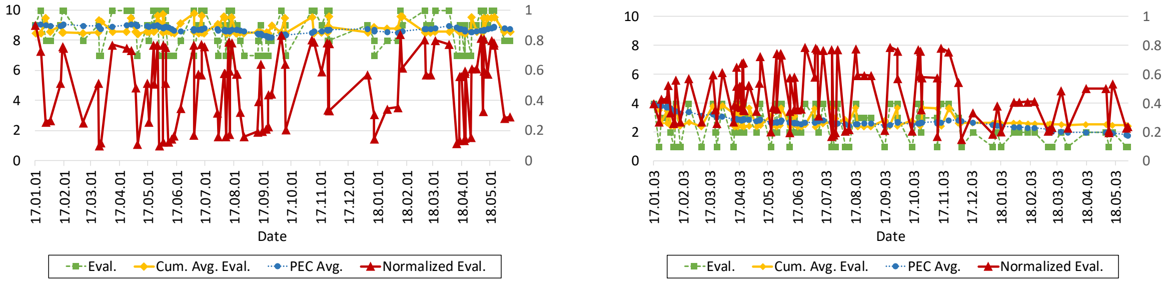
Table 1. Ten buyers' evaluation criteria.

User	EV.range
Buyer A, B	7–10
Buyer C, D	1–4
Buyer E, F	3–8
Buyer G, H	1–10
Buyer I, J	1–2 or 9–10

Table 2. Five sellers with different characteristics.

User	Characteristics
Seller A	Good
Seller B	Bad
Seller C	Balanced
Seller D	Good to Bad
Seller E	Bad to Good

Figure 8a,b plots the rating values made over time by a generous and strict buyer, respectively, together with the corresponding NE, PEC average, and naive cumulative average of rating values. Two figures demonstrate the normalization process of the rating values according to each user's grading tendency. When it comes to generous and strict users, even though raw rating values are extremely high or low, they become (almost) uniformly distributed into the range of 0 to 1 after the normalization process. On the other hand, naive cumulative average values remains either high or low, providing less meaningful information to other customers. Eventually, we can see that each user's criteria is well reflected on the NE although every user has different tendency.

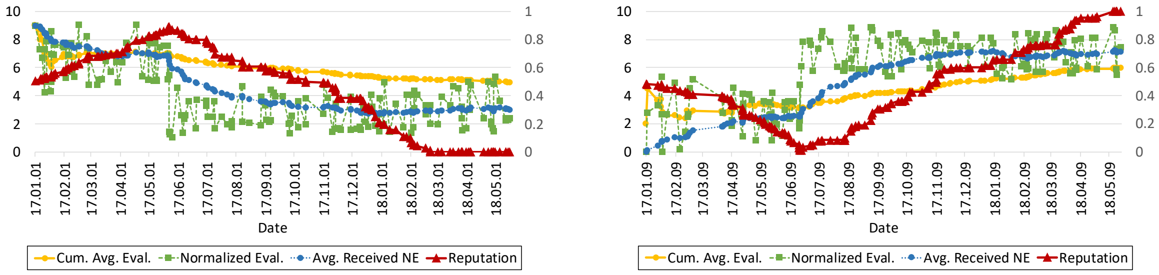


(**a**)Normalized evaluations of a generous buyer B.

(**b**)Normalized evaluations of a strict buyer E.

Figure 8. Normalized evaluations of two different buyers B and E.

Figure 9a,b plot how the reputation of "bad to good" and "good to bad" sellers change over time, according to the evaluations they receive from counterparts. The seller's reputation starts from a neutral value of 0.5 at the beginning. In Figure 9a, we observe that the reputation of seller D goes up to 0.9 approximately for six months and goes down drastically after 3 months losing all her reputation. However, naive cumulative average value does not follow that tendency well, and tends to lag behind due to large amount of out-dated history. On the other hand, seller E's reputation shown in Figure 9b goes down for the first six months, and then goes up gradually after which it starts to receive good rating values. However, again, naive cumulative average value does not follow that tendency well, and tends to lag behind due to large amount of out-dated history. These two graphs show that the model derives reputation that better reflects the positive or negative grading aspect with respect to *PEC* of an evaluator in comparison with *ARNE*.

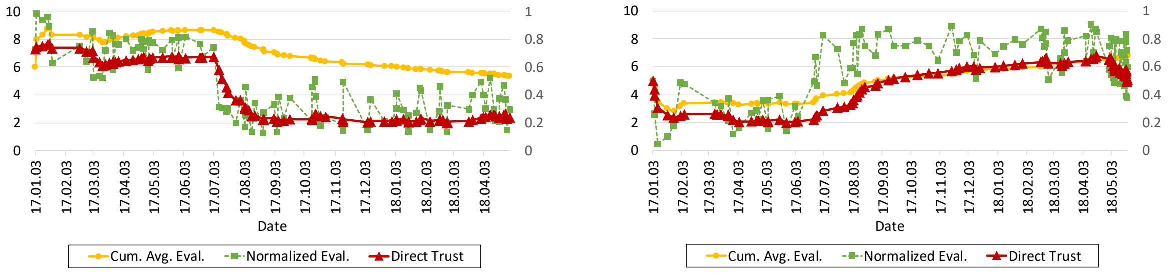


(**a**)Reputation of seller D set to change from good to bad.

(**b**)Reputation of seller E set to change from bad to good.

Figure 9. Change of reputation over time for two different sellers D and E.

Figure 10a,b plot the change of trust of buyer G to seller E and to seller D, respectively. In Figure 10a, the buyer has a strong faith at the beginning, and the trust is going up and down slightly for the next six months. Afterwards, as the buyer loses confidence in the seller, her trust drops sharply to ∼0.2. On the other hand, Figure 10b shows the buyer's trust increasing gradually from 0.2 to ∼0.6. However, for both cases, naive cumulative average value is lagging behind and does not follow the changes in rating promptly. According to these graphs, we can see that the notion "losing confidence is much easier than maintaining confidence" is well reflected, as expected.



(**a**)Trust of buyer G to seller E set to change from good to bad.

(**b**)Trust of buyer G to seller D changing from bad to good.

Figure 10. Trust of buyer G to two different sellers E and D.

Based on the above results, we have verified the effectiveness of *PEC* through which our model calculates reputation and trust values that well reflects each person's different characteristics that may change over time, and also the psychological factors we proposed in Section 3, resulting in reliable and objective metrics.

5. Evaluation via Real Bitcoin Transaction Dataset

To evaluate the effectiveness of our model on large-scale real transactions that span over long period of time, we now apply it on Bitcoin's transaction data. Among various cryptocurrencies, Bitcoin was chosen for our evaluation as it is the representative of decentralized digital currency and also has the most number of transactions from 2009 to 2018. Furthermore, Bitcoin has a simple structure, making it relatively easier to analyze its contents than Ethereum which supports more features such as smart-contract [33].

A Bitcoin transaction consists of an input section and an output section [34]. Source addresses and destination addresses are denoted in the input and output sections, respectively, and the amounts of Bitcoin the destination addresses would receive are indicated in the output section. After being verified by mining nodes, transactions are written into a block with a timestamp indicating their mined time, and the block is delivered to the neighbor nodes for network-wide propagation. Eventually, every full nodes in the world (which is around ∼8500 nodes [29]) will have the same Bitcoin ledger. Thus to conduct the experiment, it was necessary for us to dive into the sea of Bitcoin full nodes filled with undug valuable transactions.

Experiment Set-Up and Data Preprocessing: We installed the Bitcoin Core, an open-source reference software that serves as a node in the Bitcoin network (https://bitcoin.org/en/bitcoin-core/). After initialization, it started to download all the past raw blocks from the main Bitcoin network to local storage, and continued to do so as new blocks are generated continuously. We were able to collect all blocks ranging from the genesis block generated on 3 January 2009 to the one in 9 July 2018. The data consist of 531,136 blocks and 326,948,822 transactions with the size of 189 GB in total.

As depicted in Figure 11, we split the experimental set-up into two parts: the generator part and the simulator part. In the generator part, we extract and generate, from the raw transaction data of Bitcoin, meaningful data such as transaction amounts and addresses of users. BlockSci [35], an application for analyzing blockchain, is utilized for this purpose which supports three main features: parsing transactions, clustering addresses, and analyzing transactions. We first pass the raw block data into the parser to transform the data to the format used by BlockSci, which resulted in size reduction to 151GB without loss of original information.

Next step is to cluster the addresses. In Bitcoin, users can generate (Using the Bitcoin Core or any client wallet software built for Bitcoin), without limit, a number of Bitcoin *addresses* that serve as an account number (To be precise, there is no notion of accounts in Bitcoin. By account, we mean the value of the UTXOs for their corresponding keys, and those keys act like account numbers.). Thus, multiple addresses may belong to (and be used by) a person or a team on the Bitcoin's network, resulting in the necessity of clustering them into groups that are likely to be a single user having similar usage patterns. By inserting the transactions into BlockSci's clusterer [35], we extracted 214,219,988 clusters from a total of 947,837,576 addresses discovered in all transactions.

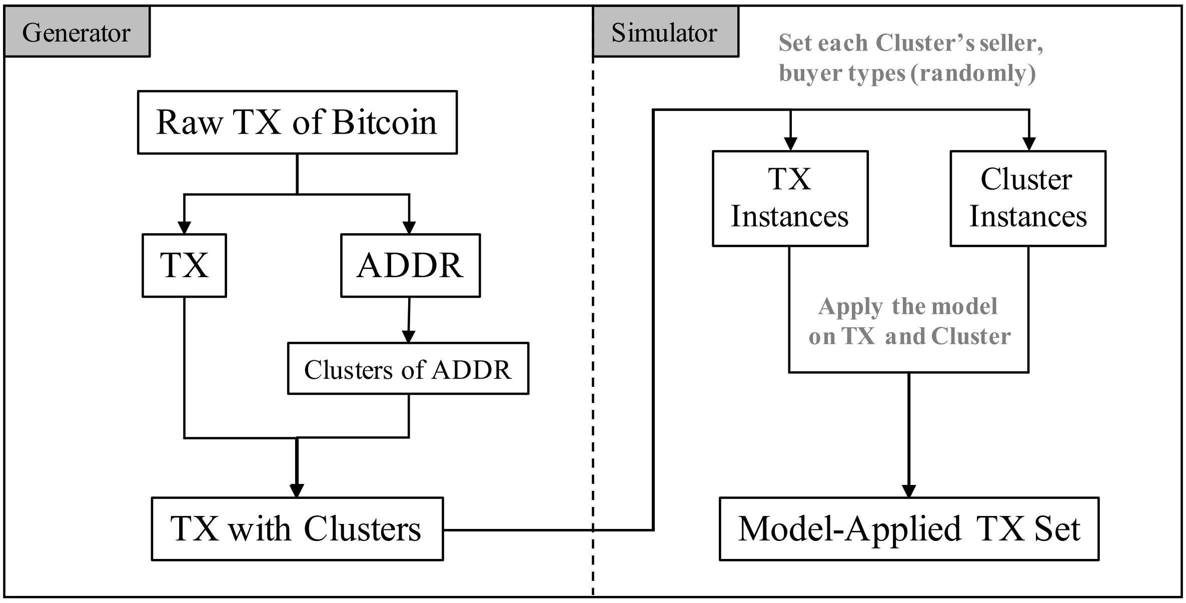


Figure 11. Flow chart on transforming Bitcoin's raw data.

After clustering the addresses, we found that there are a lot of addresses which are used only once and never again, the so-called "single-hit" addresses. They are not included in any of the clusters due to lack of any usage pattern, and these were unsuitable for evaluating our model. Thus, we then examined which clusters of addresses have done transactions similar to the ones that could happen in real-life. We counted the number of transactions occurred between every pair of clusters, and plotted the distribution of transactions among addresses as in Figure 12. It shows that most of transactions between two clusters are concentrated in the range of 1 to 10, meaning a great deal of clusters consist of a single address and they are used for once or infrequently. To avoid the single-hit clusters which are not usually discoverable in the real-world, we searched for a range of transaction frequencies within Figure 12 that includes the least number of single-hit clusters. As a result, we selected 700 clusters which have each done transactions from 100 to 200 times, and extracted a list of 215,783 transactions that involve those clusters. From those transactions, there were total of 155,040 clusters including the selected clusters. Finally, the addresses within a cluster are substituted with their unique cluster identifying number.

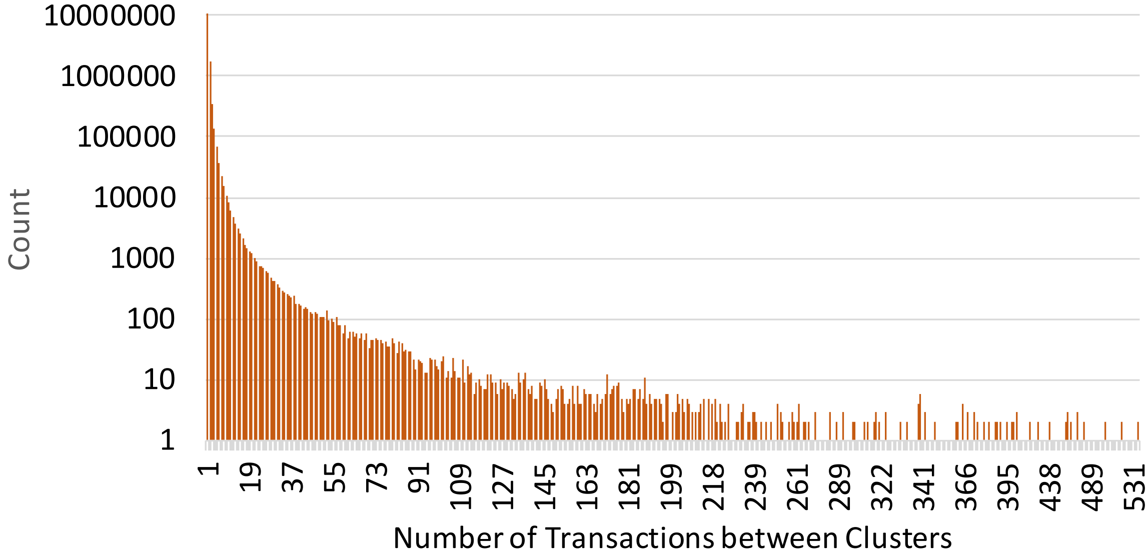


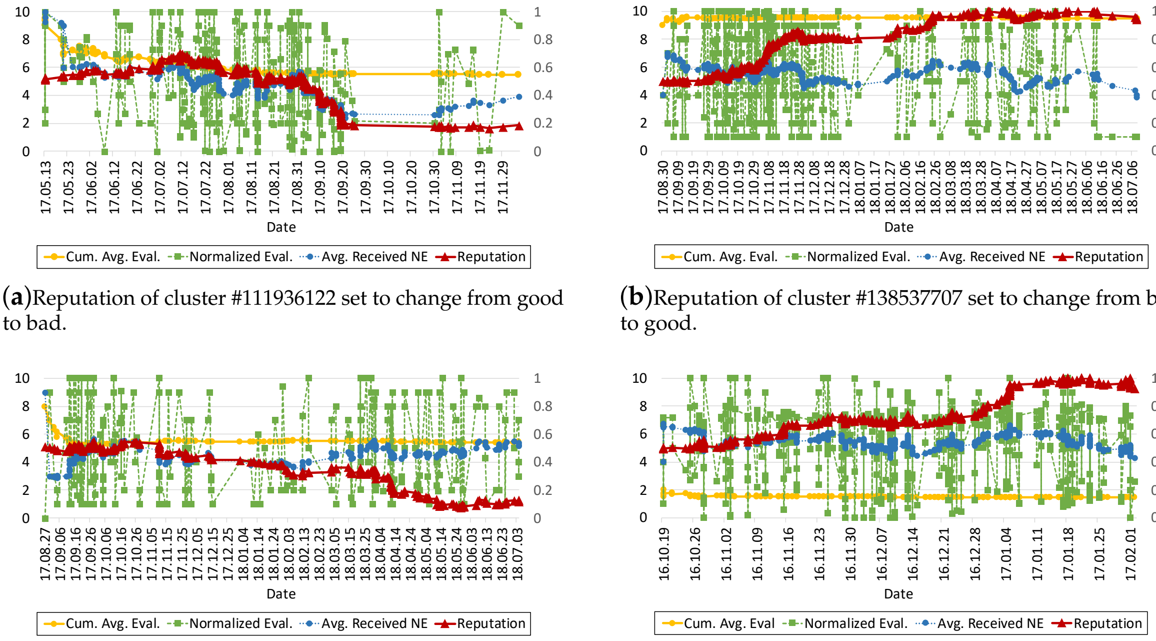
Figure 12. Distribution of transactions between addresses.

Application of the Model: So far we have extracted clusters of addresses (corresponding to users) and transactions suitable for simulating and evaluating our model. However, to apply our model on this dataset, it lacks the rating values. To resolve its shortage, we exploit the same approach used in Section 4 to generate the simulated ratings according to the criteria set in Tables 1 and 2 where the type for each cluster was selected randomly. As a final preprocessing step, our simulator sorts the transactions in ascending order based on each transaction's timestamp value.

In the simulator part (Figure 11), our system sequentially walks through the list of transactions while applying our model. It calculates the time difference (based on the timestamp) between previous and current transactions to derive *TDW*, and calculates the updated *PEC*, *NE*, and *ARNE* values. After that, those values are utilized to update the counterpart's reputation and *DTR* values. Above process is applied to both clusters of a transaction, and sequentially to all transactions in our dataset.

Analysis of Model-Applied TX Set: To analyze the result of applying our model to the transactions, we plot the reputation, *NE*, and *ARNE* of four representative clusters among 155,040 clusters as shown in Figure 13a–d.

Figure 13a presents the reputation of cluster #111936122. This cluster was selected as type "Seller D" in Table 2 which receives good ratings at the beginning for 3 months and changes to receive bad ratings for the rest of the period. The cluster has dealt with other 108 clusters 297 times for 6 months from 13 May to 6 December on the same year of 2017. Her reputation begins from 0.5 (neutral), and slightly increases for 3 months while making 65 transactions with the others. During the month of September, she loses her reputation relatively faster than the other months, from 0.5036 to 0.1848, as we intended for this seller. For this month, she receives comparatively lower *NE*s than during the other months, which are on average 0.47 and 0.31, respectively. This means that the ratings made by the numerous buyers are well reflected on her reputation, via appropriately normalized *NE*, regardless of *who* rated them (regardless of various *PEC* of the buyers).



(**a**)Reputation of cluster #111936122 set to change from good to bad.

(**b**)Reputation of cluster #138537707 set to change from bad to good.

(**c**)Reputation of cluster #137987649 set to be bad.

(**d**)Reputation of cluster #67863927 set to be good.

Figure 13. Change of reputation over time for four different sellers.

Figure 13b plots a more interesting case with cluster #138537707, which was set to treat other clusters badly for the first 4 months and to behave well afterwards. While having 273 transactions during roughly 1 year, her reputation does not go down as we expected. Investigation revealed that this unusual behavior comes from the single-hit clusters. When it comes to a user who creates an address and does a transaction with another for the first time, that rating value cannot be normalized due to lack of *PEC*. For the first 4 months during which the cluster should have received *NE*s below 0.5, it’s transactions were 189 single-hit clusters reaching up to roughly 80% of 236 transactions. As defined in Section 3, the *NE* is derived from the evaluation based on the user’s *PEC*. However, first time evaluator does not have a *PEC* defined, and if the *PEC* is not initialized, *NE* is set to the evaluation divided by 10. For instance, if Alice rates Bob as 8 for the first time on an e-commerce store, her *NE* is set to be 0.8, which affects Bob’s reputation positively despite the fact that Alice always grades between 8 and 10 and 8 is a bad rating for her. This is the sparsity problem, which could be exploited by malicious users who would like to manipulate data. However, recall that the data we use for this evaluation are extracted from Bitcoin, and single-hit addresses are a peculiarity of Bitcoin. In our real-life, there are very few users who create a new account to buy a product for every transaction. And as long as the user makes a second transaction, the model could derive desired *NE* successfully from her *PEC*.

Cluster #137987649 presented in Figure 13c has 248 transactions with 192 clusters from 27 August 27 of 2017 to 18 July of 2018. As she was had the “bad” type seller, she loses reputation gradually and continuously. On the other hand, Figure 13d presents the reputation of cluster #67863927 which the seller type is set to be good. From 19 October of 2016 to 2 February of 2017, she had transactions with 564 clusters of which the number of single-hit clusters are only 51 (9.04%). As there are few of single-hit clusters, evaluation values are properly normalized based on the concerned clusters’ *PEC* unlike the case of Figure 13b. As expected, her reputation improves as time goes by, and she maintains her reputation regardless of what type of buyers rate her.

To summarize, we exploited the experiments with the dataset generated by two distinct methods. First, we experimented with the dataset consisting of randomly generated timestamps, transactions and users including their way to grade others. In this part, the reliability and objectiveness of the model-based reputation and trust were successfully verified. Second, we extracted meaningful dataset from Bitcoin's raw transaction data by generalizing transactions and clustering addresses. By applying the model on this dataset, we were able to see the robustness of our model which could work even in a worst case by overcoming the 'sparsity' problem with our proposed idea. Consequently, the model is proven to be efficient, robust and effective in deriving trust and reputation based on each user's evaluating tendency.

6. Conclusions

In this paper, we proposed an efficient method for deriving reputation and trust from raw evaluations stored in transactions of a blockchain-based online payment system. In order to make the values more robust, reliable and objective, we applied several concepts and notions related to human behavior and psychological factors such as time difference weight, personal evaluation criteria, more belief in a person with higher reputation, friend-of-a-friend, and losing confidence, is much easier than maintaining it. The evaluation results, experimented with a simulation dataset and a Bitcoin transaction set, show the effectiveness and robustness of our model.

The model can be applied to various domains; for example, it can be used to derive trustworthiness of merchant delivery process (e.g., Craigslist) implemented on a private blockchain such as Hyperledger. Our model has been implemented and incorporated into a live commercial blockchain service platform (http://nodehome.io), and new application developments are underway to utilize the technology. As a future work, we plan to design and implement a model which computes, in real-time, reliability of each node on peer-to-peer networks in order to improve speed and performance, based on our original model.

Author Contributions: Conceptualization, J.P.; methodology, M.P. and J.P.; software, J.A. and M.P.; validation, J.A., M.P., and H.S.; formal analysis, H.S. and J.P.; investigation, M.P. and H.S.; resources, J.P.; data curation, J.A. and H.S.; writing—original draft preparation, J.A. and M.P.; writing—review and editing, H.S. and J.P.; visualization, J.A. and M.P.; supervision, project administration, and funding acquisition, J.P.

Funding: Following are results of a study on the "Leaders in INdustry University Cooperation +" Project, supported by the Ministry of Education and National Research Foundation of Korea, this research was also supported by the Chung-Ang University Research Grants in 2018.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 7 December 2019).
2. Fotiou, N.; Polyzos, G.C. Decentralized name-based security for content distribution using blockchains. In Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), San Francisco, CA, USA, 10–15 April 2016; pp. 415–420.
3. Fujimura, S.; Watanabe, H.; Nakadaira, A.; Yamada, T.; Akutsu, A.; Kishigami, J.J. BRIGHT: A concept for a decentralized rights management system based on blockchain. In Proceedings of the IEEE International Conference on Consumer Electronics-Berlin (ICCE-Berlin), Berlin, Germany, 6–9 September 2015; pp. 345–346.
4. Kishigami, J.; Fujimura, S.; Watanabe, H.; Nakadaira, A.; Akutsu, A. The blockchain-based digital content distribution system. In Proceedings of the IEEE International Conference on Big Data and Cloud Computing (BDCloud), Dalianm, China, 26–28 August 2015; pp. 187–190.
5. Kuo, T.T.; Kim, H.E.; Ohno-Machado, L. Blockchain distributed ledger technologies for biomedical and health care applications. *J. Am. Med. Inform. Assoc.* **2017**, *24*, 1211–1220. [CrossRef] [PubMed]
6. Shae, Z.; Tsai, J.J. On the Design of a Blockchain Platform for Clinical Trial and Precision Medicine. In Proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS), Atlanta, GA, USA, 5–8 June 2017; pp. 1972–1980.
7. Xia, Q.; Sifah, E.B.; Smahi, A.; Amofa, S.; Zhang, X. BBDS: Blockchain-based data sharing for electronic medical records in cloud environments. *Information* **2017**, *8*, 44. [CrossRef]

8. Yue, X.; Wang, H.; Jin, D.; Li, M.; Jiang, W. Healthcare data gateways: Found healthcare intelligence on blockchain with novel privacy risk control. *J. Med. Syst.* **2016**, *40*, 218. [CrossRef] [PubMed]
9. Cha, S.C.; Tsai, T.Y.; Peng, W.C.; Huang, T.C.; Hsu, T.Y. Privacy-aware and blockchain connected gateways for users to access legacy IoT devices. In Proceedings of the IEEE Global Conference on Consumer Electronics (GCCE), Nagoya, Japan, 24–27 October 2017; pp. 1–3.
10. Huang, Z.; Su, X.; Zhang, Y.; Shi, C.; Zhang, H.; Xie, L. A decentralized solution for IoT data trusted exchange based-on blockchain. In Proceedings of the IEEE International Conference on Computer and Communications (ICCC), Chengdu, China, 13–16 December 2017; pp. 1180–1184.
11. Novo, O. Blockchain meets IoT: An architecture for scalable access management in IoT. *IEEE Internet Things J.* **2018**, *5*, 1184–1195. [CrossRef]
12. Ouaddah, A.; Elkalam, A.A.; Ouahman, A.A. Towards a novel privacy-preserving access control model based on blockchain technology in IoT. In *Europe and MENA Cooperation Advances in Information and Communication Technologies*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 523–533.
13. Polyzos, G.C.; Fotiou, N. Blockchain-assisted Information Distribution for the Internet of Things. In Proceedings of the IEEE International Conference on Information Reuse and Integration (IRI), San Diego, CA, USA, 4–6 August 2017; pp. 75–78.
14. Cholewa, J.B.; Shanmugam, A.P. Trading Real-World Assets on Blockchain-An Application of Trust-Free Transaction Systems in the Market for Lemons. *Bus. Inf. Syst. Eng.* **2017**, *59*, 425–440.
15. Lundqvist, T.; De Blanche, A.; Andersson, H.R.H. Thing-to-thing electricity micro payments using blockchain technology. In Proceedings of the 2017 Global Internet of Things Summit (GIoTS), Geneva, Switzerland, 6–9 June 2017; pp. 1–6.
16. Seo, J.; Choi, S.; Kim, M.; Han, S. A robust ensemble-based trust and reputation system against different types of intruder attacks. *Int. J. Comput. Math.* **2016**, *93*, 308–324. [CrossRef]
17. Kim, S.; Ha, W.; Seo, J.; Han, S.; Kim, M. A method of evaluating trust and reputation for online transaction. *Comput. Inform.* **2015**, *33*, 1095–1115.
18. Carboni, D. Feedback based Reputation on top of the Bitcoin Blockchain. *arXiv* **2015**, arXiv:1502.01504.
19. Schaub, A.; Bazin, R.; Hasan, O.; Brunie, L. A trustless privacy-preserving reputation system. In *IFIP Advances in Information and Communication Technology, Proceedings of the IFIP International Conference on ICT Systems Security and Privacy Protection (SEC 2016)*, Ghent, Belgium, 30 May–1 June 2016; Springer: Cham, Switzerland, 2016; Volume 471.
20. Morid, M.A.; Shajari, M. An enhanced e-commerce trust model for community based centralized systems. *Electron. Commer. Res.* **2012**, *12*, 409–427. [CrossRef]
21. Xiong, L.; Liu, L. A reputation-based trust model for peer-to-peer e-commerce communities. In Proceedings of the IEEE International Conference on E-Commerce (CEC), Newport Beach, CA, USA, 24–27 June 2003; pp. 275–284.
22. Resnick, P.; Zeckhauser, R. Trust among strangers in Internet transactions: Empirical analysis of eBay's reputation system. In *The Economics of the Internet and E-Commerce*; Emerald Group Publishing Limited: Bingley, UK, 2002; pp. 127–157.
23. Siau, K.; Shen, Z. Building customer trust in mobile commerce. *Commun. ACM* **2003**, *46*, 91–94. [CrossRef]
24. Adomavicius, G.; Tuzhilin, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.* **2005**, *6*, 734–749. [CrossRef]
25. Han, J.; Pei, J.; Kamber, M. *Data Mining: Concepts and Techniques*; Elsevier: Amsterdam, The Netherlands, 2011.
26. Lee, J.; Lee, D.; Lee, Y.C.; Hwang, W.S.; Kim, S.W. Improving the accuracy of top-N recommendation using a preference model. *Inf. Sci.* **2016**, *348*, 290–304. [CrossRef]
27. Decker, C.; Wattenhofer, R. Information propagation in the bitcoin network. In *IEEE P2P Proceedings*; IEEE: New York, NY, USA, 2013; pp. 1–10.
28. Donet, J.A.D.; Pérez-Sola, C.; Herrera-Joancomartí, J. The Bitcoin P2P Network. *Financ. Cryptogr. Data Secur. FC Work. BITCOIN WAHC* **2014**, *16*, 87–102.
29. Park, S.; Im, S.; Seol, Y.; Paek, J. Nodes in the Bitcoin Network: Comparative Measurement Study and Survey. *IEEE Access* **2019**, *7*, 57009–57022. [CrossRef]
30. Seol, Y.; Ahn, J.; Park, S.; Ji, M.; Chae, H.; Yi, J.; Kim, Y.; Paek, J. Query-chain: Fast and Flexible Blockchain-based Platform for Diverse Application Services. In Proceedings of the International Conference on Information and Communication Technology Convergence (ICTC), Jeju-si, Korea, 16–18 October 2019.

31. Ahn, J.; Park, M.; Paek, J. Reptor: A Model for Deriving Trust and Reputation on Blockchain-based Electronic Payment System. In Proceedings of the International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea, 17–19 October 2018; pp. 1431–1436.
32. Herbert, J.; Litchfield, A. A novel method for decentralised peer-to-peer software license validation using cryptocurrency blockchain technology. In Proceedings of the 38th Australasian Computer Science Conference (ACSC), Sydney, Australia, 27–30 January 2015; Volume 27, p. 30.
33. Buterin, V. Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform. 2017. Available online: https://ethereum.org (accessed on 7 December 2019).
34. Herrera-Joancomartí, J. Research and challenges on bitcoin anonymity. In *Data Privacy Management, Autonomous Spontaneous Security, and Security Assurance*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 3–16.
35. Kalodner, H.; Goldfeder, S.; Chator, A.; Möser, M.; Narayanan, A. BlockSci: Design and applications of a blockchain analysis platform. *arXiv* **2017**, arXiv:1709.02489.