IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 19 June 2024, accepted 28 July 2024, date of publication 5 August 2024, date of current version 6 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3439012

RESEARCH ARTICLE

MoMTSim: A Multi-Agent-Based Simulation Platform Calibrated for Mobile Money Transactions

DENISH AZAMUKE, MARRIETTE KATARAHWEIRE, AND ENGINEER BAINOMUGISHA
Department of Computer Science, Makerere University, Kampala, Uganda

Corresponding author: Denish Azamuke (azamuke.denish@students.mak.ac.ug)

This work was supported in part by the Digital Credit Observatory (DCO), a program of the Center for Effective Global Action (CEGA), with support from the Bill & Melinda Gates Foundation; JPMorgan Chase & Company; and Google Ph.D. Fellowship Program.

ABSTRACT Research on multi-agent systems has extensively modeled real-world phenomena across various domains including epidemiology, urban planning, and financial transactions. These systems often struggle to produce agent behaviors that comprehensively capture the dynamics of the real-world ecosystem and the unique behaviors of each agent type. Furthermore, the limited explainability of these models due to non-iterative calibration poses significant challenges. This paper introduces an iterative model calibration algorithm that dynamically adjusts the multitude of parameters in a multi-agent simulation platform. Initially treating the simulation model as a black box, our method refines simulation parameters through cycles of adjustments based on clusters of observed behaviors comprising the behavior of both agents and actors. This approach allows for the identification and correction of inaccuracies, introduces new parameters, and discards erroneous ones within the agent-based model as demonstrated in a **Mo**bile **M**oney **T**ransaction **Sim**ulator (**MoMTSim**). The calibration algorithm enhances the realism and applicability of the simulation model by ensuring that the generated synthetic datasets closely mirror real transaction data. The effectiveness of this calibration method was determined by validating the generated data through comparing the real and synthetic datasets using statistical methods including the Kolmogorov-Smirnov test, the sum of squared errors (SSE) method, and Bland-Altman plots. We computed the delta between the real and synthetic data using the SSE approach and found that the synthetic datasets resemble real data. This shows that MoMTSim effectively generates synthetic data that closely matches real mobile money transaction data, validating the accuracy of our model calibration algorithm in simulating complex financial ecosystems.

INDEX TERMS Multi-agent-based simulation, model calibration, synthetic data, mobile money.

I. INTRODUCTION

Research on calibrating multi-agent systems to model complex scenarios of real-world ecosystems, epidemiology, urban planning, and financial transactions, continues to advance. Many of the studies have focused on the modeling of the multi-agent systems including epidemics [1] and pandemics such as COVID-19 [2], urban form [3], climate scenarios [4]. Similarly, agricultural management scenarios [5], drone strikes and radicalization [6], micro-level dynamics of water reuse [7], traffic, and disaster response [8] have been modeled. Social networks created due to financial transactions have also been studied using multi-agent-based simulation (MABS) of the actions of entities involved in the transactions [9], [10], [11]. However, it is notoriously difficult to ensure that agents in the multi-agent systems produce realistic behaviors that cover in detail, characteristics of the real ecosystem and also represent the unique attributes and operations of each agent type [12] without the calibration of the numerous parameters of a simulation model. Several studies achieved the simulation of credible agents including common scenarios of disease spread [1], [2], [8] based on data

The associate editor coordinating the review of this manuscript and approving it for publication was Derek Abbott.

on properties of real actors. Nevertheless, without explicitly calibrating simulation parameters using a combination of simulation logs and real data, the explainability of such models is limited [12] and this hinders their usage. Besides the missing behaviors of agents, lacking interactions and errors in simulation can hardly be traced when limited, non-iterative approaches are employed for validating simulations in multi-agent systems [12], [13].

The core contribution of this paper is the development and implementation of an iterative model calibration algorithm that is used to dynamically calibrate several parameters of multi-agent systems. Our algorithm begins by treating the multi-agent simulation model as a black box, progressively refining the simulation parameters through cycles of adjustment based on observed behaviors of agents and properties of real transaction data. This method not only corrects inaccuracies but also adapts the simulation to incorporate emergent behaviors and interactions, a significant advancement over static, non-iterative calibration methods [14], [15], [16] that are limited by predefined equations and models. Thus, the iterations in this approach allow for the discovery of missing behaviors and the introduction of new parameters in the simulation model as well as the removal of errors.

The effectiveness of our approach has been demonstrated through the calibration of the multi-agent mobile money financial system, whereby the generated synthetic datasets were validated against real financial data using statistical methods such as the sum of squared errors (SSE), the Kolmogorov-Smirnov (KS) test, and Bland-Altman plots. This validation confirms that our calibrated simulation model produces data that closely resembles real financial transactions, enhancing the realism and applicability of the algorithm in calibrating an agent-based system.

In this study, we focus on the mobile money domain to demonstrate the utility of the calibration algorithm, leveraging a multi-agent simulation platform MoMTSim [17]. MoMTSim models mobile money transactions representing transactions in the real world that are carried out via mobile devices without the need for traditional banking infrastructure. MoMTSim simulates a spectrum of mobile money transactions such as deposits, withdrawals, transfers, debits, and payments, reflecting the complex interactions within a typical financial ecosystem. Agents in the simulation platform represent customers, banks, and merchants and they possess unique attributes derived from real-world data, ensuring an accurate and detailed representation of the real financial ecosystem. This paper makes the following contributions:

- We introduce an iterative algorithm that dynamically calibrates multi-agent systems based on simulation observations and real-world data, significantly improving the accuracy and realism of the models.
- We apply our model calibration approach to mobile money transactions using the MoMTSim platform, which realistically models the intricate dynamics of the financial ecosystem and transactions conducted via mobile devices.
- This study validates the efficacy of our model calibration approach by comparing the synthetic datasets generated from the calibrated MoMTSim model with actual mobile money transaction data using statistical methods, including the KS test, SSE method, and Bland-Altman plots, thereby confirming the enhanced realism of the MoMTSim simulations.

The rest of the paper is organized as follows: section II describes related work on model calibration. Section III presents the modeling and calibration algorithm for a multi-agent system and demonstrates its usage in the calibration of MoMTSim. Section IV discusses the results of model calibration for a multi-agent mobile money financial simulation platform and conclusions are made in section V.

II. RELATED WORK

Singh et al. [1] use survey data to aid the calibration of a large multi-agent simulation for epidemics. The study applied a Markov Decision Process (MDP) to model agent decision-making regarding disease avoidance behaviors. Each behavior (such as washing hands or using hand sanitizers) was mapped to interventions in the simulation that could affect the susceptibility of an agent to influenza. Parameters of the MDP, including the cost associated with each behavior, were adjusted. The costs influenced how likely an agent was to adopt a behavior, balancing the cost against the benefit of reducing infection risk. Parameters of the model were iteratively tuned using optimization algorithms such as Numerical Gradient Descent (NGD), Cross Entropy (CE), and Smoothed-Cross Entropy (SCE) methods to minimize the difference between the modeled behaviors and those observed in the survey. Outputs from the calibrated epidemic model were validated against the survey data to ensure that the distribution of behaviors matched those observed during real-world influenza outbreaks.

Züfle et al. [3] introduces the Urban Life agent-based simulation which was calibrated by adjusting parameters to ensure that the outputs of the simulator align with observed or expected patterns in real urban settings. The study relied on these observations to assess the conformity of the agent model to the real world. In the model, agents meet in places forming a social network. An infectious disease modeling was carried out in the study to predict the spread of a hypothetical disease and prescribe measures to mitigate the spread. The model was implemented using the MASON toolkit which is core in Java and has a GIS extension, GeoMASON [18].

Another study [4] adopted a two-step process for calibrating the WRAP (water reuse adoption by farmers) model. Code walkthroughs and testing the effects of input parameters on the model results were used to verify the model. Then sensitivity analysis involved using a fractional factorial design to investigate the significance of the daily flow rates of wastewater treatment plants, the availability of primary water sources, and the pricing of recycled water factors and their interactions, focusing on how they affected the total recycled water consumption of farmers.

Groves-Kirkby et al. [2] calibrates an agent-based model for the COVID-19 pandemic scenarios to support healthcare planning. The calibration method involved adjusting model parameters to match hospital admissions, intensive care occupancy, and deaths due to COVID-19 until the model output closely resembles the three daily data streams. The study demonstrates that the calibrated model was able to provide acceptable fits to unseen data streams, including official estimates of COVID-19 incidence.

Ullah et al. [5] model agricultural management scenarios encompassing business as usual (BaU) and a climate-smart (no-till) practice using an agent-based modeling approach. A combination of geospatial data, historical crop data, and socio-economic factors was used in the model calibration. The adoption of carinata by farmers was modeled as a function of both economic profitability and social influence from neighboring farmers. Model parameters were iteratively adjusted to align the model outputs with observed data and literature regarding farmer behavior and economic conditions.

Shapiro and Crooks [6] set initial conditions and parameters for a simulation model informed by empirical data on drone strikes and terrorist attacks. The study relied on data from the Bureau of Investigative Journalism and the National Consortium for the Study of Terrorism and Responses to Terrorism (START). Key parameters such as the rate of drone strikes, the likelihood of carrying out terrorist attacks following a strike, and the distributed lag between drone strikes and resulting terrorist activities were adjusted based on parametric explorations. Therefore, the relationship between the frequency of drone strikes and the resulting terrorist attacks was calibrated against observed data to ensure that the simulated outcomes aligned with real-world data. The model outputs were cross-validated using unseen data to validate the model.

The calibration of the PaySim [19] model involved model verification where behaviors are inspected and model validation where the error rates for the real and synthetic datasets are compared using the SSE approach. PaySim is a financial simulator for mobile money transactions, designed following the MABS approach. Its initial version was not calibrated using real mobile money transaction data. PaySim was revised [9], [10] and calibrated using a sample of real mobile money transaction data with a focus on financial fraud detection. The validation of synthetic output in PaySim and the validation strategy used in our work are closely related, however, this study uses an iterative approach for the model calibration which allows for the identification of missing behaviors based on a behavioral questionnaire, the introduction of new parameters, and systematic removal of errors among other things.

Related studies mainly cover the modeling of artificial ecosystems using an agent-based modeling approach. However, they highlight the need for efficient calibration of model parameters to explain agent interactions, missing behaviors, and errors in the system. These studies lack iterative algorithms that dynamically calibrate the numerous parameters of the simulation model. Our approach enhances the state-of-the-art in calibrating multi-agent systems. We provide insights into model calibration using an iterative algorithm that has been demonstrated to calibrate an artificial financial system with many parameters for generating realistic synthetic datasets.

III. MODELING AND CALIBRATION

Several multi-agent platforms for either general purpose or domain-specific purpose exist in literature [20]. This study used MASON [21] to model MoMTSim [17]. MASON is a comprehensive and validated MABS toolkit with extensive features. It is generic and can be used for modeling any scenario of an artificial world. We found MASON suitable for our use case because it supports parallel processing, enhancing its ability to manage computationally demanding simulations more effectively than other simulation toolkits such as NetLogo [22], RePast [23] and AnyLogic [24].

MoMTSim has a client agent with a profile and the simulation model has other files including initial balance distributions, types of transactions, overdraft limits for clients, and the maximum number of occurrences per client among other things. These files form the input data for a simulation [9], [10] in MoMTSim. Similar to the real ecosystem, clients carry out transactions including deposits, transfers, debits, payments, and withdrawals, which are recorded within the system. Client involvement in a transaction is determined by a random variable, which is dependent on probabilities calculated from the analysis of real mobile money data [9], [10]. To arrive at the random variable for determining client participation in a transaction, probabilities were first computed by analyzing real mobile money transaction data. This analysis involved examining patterns and frequencies of client transactions in the historical mobile money financial data. Based on this analysis, probabilities were assigned to different scenarios and conditions under which clients typically participate in financial transactions. Thus the probabilities were used to define the random variable, which dictates the likelihood of the participation of a client in any given transaction in the simulation model. The random variable was generated in a way that mirrors the observed behaviors in the real mobile money financial data. Consequently, the state of the client can be modified in the simulations.

A. SIMULATION MODEL

1) SCENARIO-BASED ANALYSIS

Simulating the mobile money service with more detail requires the simulation of the various types of transactions that are present in the real ecosystem.

Transactions were grouped by similarity based on their characteristics forming scenarios. The payment of domestic bills, and purchase of airtime, goods, and services fall under payment transaction type. Whenever a mobile money user puts money in their account via a mobile money merchant, it is a deposit, while withdrawal involves a user removing e-money off their account as hard cash via a mobile money merchant. The intermediate movement of money in the system by users such as a client sending money to another client is transfer. With online and mobile banking services in the Sub-Saharan region, users often send money to bank accounts from mobile money accounts and this type of transaction is debit.

Some transactions involve a user of a mobile money service and a non-user commonly referred to as a voucher that is often created and then redeemed. Credit is the opposite of a debit transaction and saving money on the mobile money platform is a transaction aimed at promoting savings among the masses. Due to the insignificance of savings, vouchers, and credit transactions in the real ecosystem including in the sample of the real transaction data used for statistical analysis, we decided not to include them in the artificial financial ecosystem. Regular users who participate in major transactions such as deposits, withdrawals, transfers, payments, and debits are often recorded significantly in the real transaction data. These users are usually the ones who can save money on the platform. They also participate in other minor transactions including savings, vouchers, and credits. An agent-based financial platform with the major transaction types would sufficiently output datasets that represent attributes of the real ecosystem. A summary of the mobile money transaction types with their significance and transaction rules is presented in Table 1.

2) STATISTICAL ANALYSIS

Real transaction data includes transaction amounts associated with transaction types, transaction IDs for the initiator and recipient of the transaction, and the initial and final balance for the initiator and recipient of a transaction, among other things. The count statistics of the real data were computed to determine the frequency of the transactions over time using Poisson regression [25]. Poisson regression supports the modeling of the count of transactions that happen at a consistent rate within a fixed interval of time. It is given by (1)

$$\lambda_i = e^{(\beta_0+\beta_1 X_{1i}+\ldots+\beta_k X_{ki})} \tag{1}$$

where λ_i is the expected count of transactions for unit i and $X_{1i}, \ldots, X_{ki}$ are the explanatory variables.

While aggregating the transaction amounts, summary statistics including the sum, mean, and totals across different categories (per user and period) were determined. The mean is given by (2)

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{2}$$

for the individual transaction amounts x_i and the number of transactions n. The standard deviation σ for the aggregated transaction amount was determined using (3)

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i-\bar{x})^2}{n-1}}. \tag{3}$$

For the initial balance distributions for clients, the continuous data including the account balance in the real data were categorized into discrete ranges (bins). Supposing that X is the balance that we want to bin into intervals of width w. The bin for a given balance x is determined using (4)

$$\text{Bin}(x) = \left\lfloor \frac{x-\min(X)}{w} \right\rfloor \tag{4}$$

where $\min(X)$ is the minimum balance in the dataset. It follows that the counting of the number of data points that fall into each bin is given as f_i = count of X such that $\text{min\_range}_i \leq X \ \text{max\_range}_i$ for a given bin i and frequency f_i.

Thus, the percentage p_i is calculated using (5)

$$p_i = \left(\frac{f_i}{N}\right) \times 100 \tag{5}$$

where N is the total number of observations. The client profiles were analyzed using cluster analysis [26] to segment clients into groups with similar transaction behaviors. This was achieved using K-means clustering, as shown in the flowchart in Fig. 1. Sequential steps involved in clustering a batch of real mobile money transactions to extract client profiles are presented in the flowchart. Starting from the data collection phase, where transaction data were gathered from financial institutions from Sub-Saharan Africa, followed by data cleaning and preprocessing, which ensures data quality and usability. Features in the data were engineered and selected to identify and refine features that effectively represent transaction behaviors including the transaction frequency, amount, and location of the user.

After preparing the data and standardizing it, K-means clustering is initialized. The clustering loop involves assigning transactions to clusters and updating cluster centroids iteratively until convergence is achieved.

Post-clustering, the clusters undergo a series of validations including computing the Silhouette score and calculating the Davies-Bouldin Index to assess the quality and distinctiveness of the clusters. Visual Assessments were performed to further confirm the clustering results. Depending on these assessments, a decision is made (Re-cluster Needed?) on whether additional clustering iterations are required.

Once validated, the clusters are used for analysis and profiling, leading to the reporting and utilization phase. The derived client profiles were employed as inputs for MoMTSim. This supports the realism and effectiveness of the simulation model in MoMTSim. This marked the end of the K-means clustering for the generation of the client profiles from the real data as presented in Fig. 1.

TABLE 1. Transaction types in the real mobile money financial ecosystem.

Transaction type	Transaction rules and description	Significance
Deposit	This is concerned with a mobile money user putting money in their account via a mobile money merchant. It increases the balance of the mobile money account for the user.	Major
Withdrawal	This involves removing money from a mobile money account with the help of a mobile money merchant and it decreases the account balance of the user.	Major
Debit	Debit is the transaction delivered to a bank account from a mobile money account. It increases the bank account balance while it decreases the balance of the mobile money account that initiates the transaction.	Major
Transfer	Transfer transaction involves sending money to another user within the mobile money system. In this case, the account balance for the initiator of the transaction decreases while the balance for the recipient increases.	Major
Payment	This is concerned with purchases made for goods and services using mobile money technology. The account for the seller of the good or service increases while the account balance for the recipient of the good or service decreases.	Major
Saving	Saving refers to keeping money in a mobile money wallet (m-wallet) for future use. It increases the balance of the m-wallet.	Minor
Voucher	This is a transaction involving a user of a mobile money service and a non-user redeeming money from the service via a mobile money merchant. It reduces the account balance of the mobile money account.	Minor
Credit	Credit involves receiving money from a bank account on a mobile money account. It is the opposite of a debit transaction, the mobile money balance increases while the bank account balance decreases.	Minor

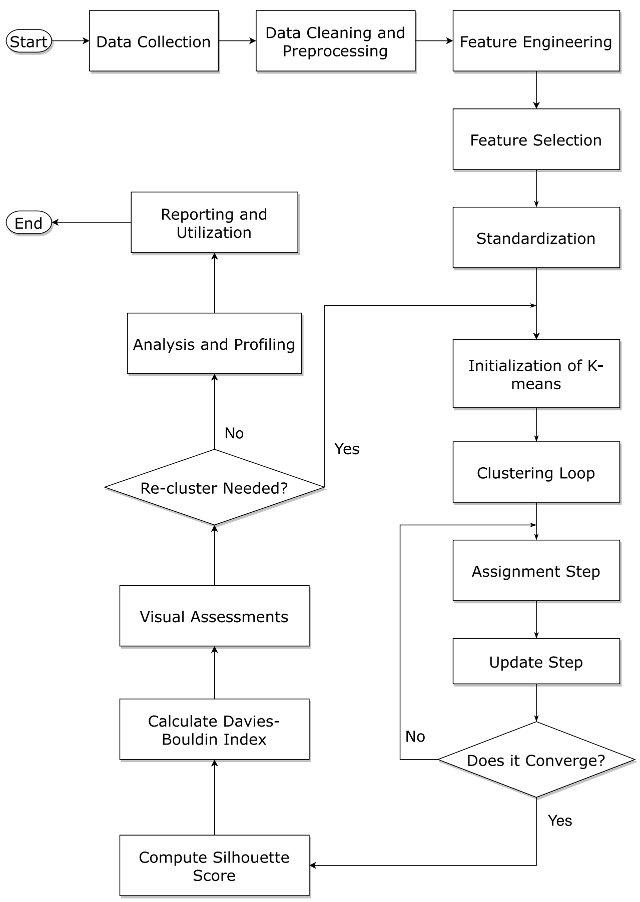


FIGURE 1. Flowchart for clustering mobile money transaction data.

3) SOCIAL NETWORK ANALYSIS

The interactions between users of mobile money services form a social network, and social network analysis [27] enabled us to uncover patterns of money flows across the network. The actors including clients, mobile money merchants, and banks, and their attributes form the nodes while the transaction types form the edges of the network graph. The degree of centrality of nodes determines the number of direct connections each node has, indicating active users of mobile money services. The degree of centrality is given by (6)

$$C_D(i) = \frac{\text{Degree of } i}{n - i} \tag{6}$$

for a node i and the total number of nodes in the social network graph n. While betweenness centrality establishes the frequency at which a node appears on the shortest paths between other nodes in the network. For mobile money services, nodes with high closeness centrality can execute transactions quickly.

The design of the simulation platform is based on the key processes of the real financial ecosystem. The simulation model relies on aggregated mobile money transaction data and transaction rules to schedule agents to participate in transactions during a simulation. The interactions of the agents in the virtual ecosystem produce synthetic output logs as shown in Fig. 2. At this point, the simulation platform must be calibrated to enforce realism. Once optimal parameters are achieved, the resulting synthetic datasets from the simulation platform are validated using the SSE method, KS test, and Bland-Altman plots. The SSE method computes the difference between the real and synthetic data. SSE is given by (7)

$$\text{SSE} = \sum_{1}^{n} (y_i - \hat{y}_i)^2 \tag{7}$$

where n is the total number of data points and y_i is the observed value for the ith data point. $\hat{y}_i$ is the model-generated prediction for the ith data point. Therefore, the SSE approach quantifies the deviation of the predictions of the simulation model from the observed data across all points.

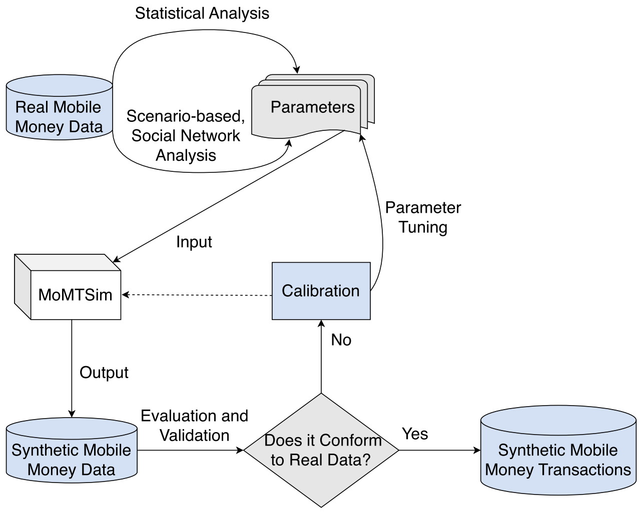


FIGURE 2. High-level view of the MABS model for mobile money transactions.

We computed the KS statistics to compare the distributions of the real and synthetic mobile money transaction datasets. KS statistic D quantifies the maximum distance between the cumulative distribution functions (CDFs) of the two transaction datasets. It is given by (8)

$$D = \sup_x |\mathrm{F}_{1,n}(x) - \mathrm{F}_{2,m}(x)| \tag{8}$$

where $\mathrm{F}_{1,n}(x)$ and $\mathrm{F}_{2,m}(x)$ are the empirical distribution functions of the real and synthetic datasets, respectively, n and m are the number of observations in each dataset, sup denotes the supremum (maximum difference across all values of x). The significance of the KS statistic was assessed by the p-value, which helps determine whether the observed differences in distributions could occur by chance under the null hypothesis. A high p-value (typically greater than 0.05) indicates a lack of evidence to reject the null hypothesis, suggesting that the two datasets could be from the same distribution.

To validate the agreement between the real and synthetic data, we visualized the transaction count, total transaction value, and average transaction value using the Bland-Altman plot. Bland-Altman plot determines how well the synthetic dataset approximates real data on the transaction count, total transaction value, and average transaction value metrics thus identifying any biases and inconsistencies in the synthetic data generation process. It follows that for each paired measurement of real data x_i and synthetic data y_i for transaction count, total transaction value, and average transaction value metrics, the differences d_i given by (9)

$$d_i = x_i - y_i \tag{9}$$

and averages a_i given by (10)

$$a_i = \frac{x_i + y_i}{2} \tag{10}$$

are determined and they form the basis for plotting and analyzing the agreement between datasets. Each difference d_i is plotted against its corresponding average a_i. The mean difference $\bar{d}$ is calculated as the average of all differences, given by (11)

$$\bar{d} = \frac{1}{n}\sum_{i=1}^{n} d_i \tag{11}$$

and is plotted as a horizontal line across the Bland-Altman plot, indicating the central tendency of the differences. The limits of agreement are calculated to determine the range within which most differences are expected to lie. Assuming that the differences are normally distributed, it follows that the standard deviation of the differences SD_d is (12)

$$\mathrm{SD}_d = \sqrt{\frac{1}{n-1}(d_i - \bar{d})^2} \tag{12}$$

whereby the limits are defined as $\bar{d} \pm 1.96 \times \mathrm{SD}_d$ and are added to the Bland-Altman plot as two horizontal lines, framing the mean difference line.

B. MODEL PARAMETERS

Simulations are managed using various parameters such as simulation configuration parameters, entity parameters, and fraud probability parameters among other things which are described as follows.

1) SIMULATION CONFIGURATION PARAMETERS

These are variables and settings that define the operations and behaviors of the simulation platform. The simulation configuration parameters include the number of steps, the seed, and a scale factor (multiplier).

- Number of steps. A step maps to an hour in the real world. Therefore, the total number of 720 steps in the artificial world represents 30 days of transactions, implying a month of activity in the real world.
- Multiplier. This is a scale factor for amplifying or reducing numerical values including amounts of transactions during a simulation.
- Seed. Simulations are random and the seed is a foundational parameter that influences the initial state and subsequent progressions of simulations. It primarily governs the reproducibility and randomness of simulations.

2) ENTITY PARAMETERS

The entity parameters are the number of agents set for a given simulation. They include the number of banks, clients, fraudsters, and mobile money merchants in the simulation environment. The number of agents enables the specification of a definite population of agents in the artificial world, similar to the real financial ecosystem. The fraudster is either a client or a mobile money merchant (can at times be the bank) exhibiting financially dishonest behavior. For every agent, the total number is set for a given simulation before it commences. Other agent metrics can be introduced in the artificial financial ecosystem using an entity-specific parameter based on the properties of actors in the real-world financial ecosystem.

3) FRAUD PROBABILITY PARAMETERS

These parameters are specifically formulated for assigning probabilities for agents designed to act fraudulently so that they can participate in transactions. Fraud probability parameters were determined by analyzing historical mobile money transaction data to identify patterns and behaviors indicative of fraud. Key fraud indicators such as abnormal transaction frequencies and amounts were identified through this analysis. These indicators were then statistically modeled to assign probabilities to each parameter, quantifying the likelihood of fraud. These probabilities were aggregated to create an overall fraud probability score for each transaction or actor involved. The fraud probability parameters in the simulation model include:

- The likelihood of a fraudulent client to commit fraud.
- The probability of a fraudulent client using a previous mobile money merchant to commit fraud. In the real ecosystem, legitimate clients have favorite mobile money merchants for carrying out transactions. Similarly, fraudulent clients often look out for their favorite mobile money merchants.
- The likelihood of a fraudulent client targeting new victims for fraud.
- The probability of mobile money merchants being at risk for fraud.

4) TRANSACTIONAL AND FINANCIAL CONSTRAINTS

These include limits on transactions such as the maximum transaction limit for transfers in mobile money among other things. Transactional and financial constraints are used for assessing the efficacy of common financial crime control mechanisms in the real ecosystem. These can be adjusted based on financial regulation policies in the real ecosystem.

5) INPUT DATA FILES

Essential data from real transactions is needed for the simulations to happen and the input data files have paths specified to CSV files containing the types of transactions, client profiles, initial balance distributions, overdraft limits for clients, and the maximum number of occurrences per client.

- Types of transactions. These are transactions to be simulated including deposits, transfers, debits, payments, and withdrawals.
- Client profiles. They contain the transaction behavior for clients.
- Initial balance distributions. This file describes the initial balance for entities (clients) within the simulation environment. This file mimics the initial balance for clients in the real financial ecosystem.
- Overdraft limits for clients. These limits influence the validity of transactions.
- Maximum number of occurrences per client. These limits determine the maximum number of times a client is involved in transactions during a simulation.

6) SIMULATION OUTPUT AND FRAUD PATTERN PARAMETERS

Simulation output parameters include paths to output files while fraud pattern parameters are meta-data on fraud typologies.

- Folder for fraud typologies. This folder contains data on the properties of various fraud scenarios. It is necessary for simulating unique fraudulent behaviors.
- Path for output files. Once the simulation has been completed, the output files are stored in this specified directory.

7) DATABASE CONFIGURATIONS

The database configurations provide an alternative for saving synthetic files to a database, especially for easy querying during analysis [9], [10]. All parameters are useful for refining and tailoring the simulation process to accurately mirror real-world situations. Additionally, they enable the expansion of simulations to encompass a vast number of agents and transactions. A successful simulation in MoMTSim undergoes the following processes.

- Loading of input files, assignment of initial balance for every client, and mobile money merchants are set up so that clients and mobile money merchants can participate in transactions during the simulation.
- MoMTSim converts a simulation into a day/hour pairing once input files have been loaded. In the simulation, a probability is assigned to each transaction. This probability is integrated into the client's model [9], [10], making the client aware of several key factors: the number of transactions they are involved in, their participation in subsequent phases of the simulation, the probability distribution for controlling their engagement in specific transactions, and their initial account balance [19]. Thus, MoMTSim models the stochastic processes based on Markov processes, strictly adhering to probabilistic rules extracted from real mobile money transaction data. This methodological foundation allows the simulation to accurately replicate the dynamics observed in the real mobile money ecosystem. A client that has been generated carries transactions with other clients based on the loaded distributions.
- Saving of log files, CSV files of raw mobile money transactions, and aggregated transactions that resemble one for the real data together with parameter file history finalize the simulation process [9], [10].

Calibration of MoMTSim encompasses the verification of agent behavior so that it reflects the behaviors of actors in the real ecosystem. This is possible by tuning the parameters of the simulation platform and the validation of the synthetic

Algorithm 1 Model Calibration Algorithm

Input: Initial parameters $P = \{p_1, \ldots, p_i\}$, transaction data, questionnaire results.
Output: Calibrated parameters for MoMTSim.

Initialization:
Cluster behaviors:
$\mathcal{C}_M$: Shared by agents and actors.
$\mathcal{C}_a$: Agent errors, removed.
$\mathcal{C}_A$: Actor misses, added.

Model Verification:
Analyze logs, data, and transaction history.
Use questionnaires to verify agent behaviors.

Calibration Iteration:
for each agent behavior **do**
  Preserve $\mathcal{C}_M$, remove $\mathcal{C}_a$, adjust for $\mathcal{C}_A$
end for
Validate global behavior accuracy.

Parameter Set Formulation:
Generate new sets:
Select $\mathcal{P}(a_i) = P_i$ where $p(P_i) = n \cdot \frac{p(b)}{|P_j|}$

Parameter Space Exploration:
Sample new sets $P_k \notin \mathcal{P}$, include if valid.

Iterative Calibration:
Refine model, incorporate new data.

Validation:
Compare with real data, adjust as needed.

Documentation:
Record changes, update MoMTSim parameters.

datasets to ensure that they are as close as possible to the real data.

C. MODEL VERIFICATION AND CALIBRATION

In this study, model verification is an iterative process leveraging documented properties of mobile money transactions, expert opinions, transaction patterns in the real data, and our understanding of the real mobile money financial ecosystem. We present a calibration algorithm for an agent-based model with the logic shown by Algorithm 1. Analysis of the transaction logs, aggregated transactions, and parameter file history form a critical part of model verification. In support of the model verification process, we used a behavioral questionnaire from a survey to analyze the components of the simulation model and elucidate agent behavior. Clusters of behaviors were formulated out of the properties of agents and actors. Actors are real-world entities in the mobile money ecosystem while agents are their representatives in the artificial financial ecosystem. The clusters formulated include.

- A cluster of behaviors associated with both agents and actors $\mathcal{C}_M$.
- A cluster of behaviors exhibited by only agents $\mathcal{C}_a$.
- A cluster of behaviors exhibited by only actors $\mathcal{C}_A$.

$\mathcal{C}_M$ denotes clusters that are correctly reproduced in the simulation. Whereas $\mathcal{C}_a$ are errors in the simulation whose parameter sets should be removed while $\mathcal{C}_A$ are missing behaviors in the simulation platform that are injected in subsequent iterations.

1) CALIBRATING THE AGENT MODEL

The agent model in MoMTSim was initially considered a black box implying it can produce different behaviors based on the initial parameters. Therefore, model calibration ensures that every agent's behavior is believable [12], [13] based on documented properties of mobile money transactions and attributes of the real financial ecosystem. This implies that the parameter set $P = \{p_1, \ldots, p_i\}$ where i represents the number of parameters is individually valid. On execution of one iteration of the simulation, valid parameters leading to behaviors $\mathcal{C}_M$ were preserved. Parameters for $\mathcal{C}_a$ and $\mathcal{C}_A$ were further examined in the next iterations. Besides, calibration guarantees that the distribution of parameter sets $\mathcal{P} = \{P_1, \ldots, P_n\}$ with n representing the number of agents globally reproduces the behaviors of actors in the real ecosystem [12], [13]. Thus, we removed parameter sets for $\mathcal{C}_a$ from the group of valid parameter sets since they did not reflect any behaviors of real-world actors.

2) FORMULATING NEW PARAMETER SET

New agent behaviors were generated by defining new parameter sets after the first iteration. This implies we define valid parameter sets $\mathcal{P}_v$ related to valid behaviors $\mathcal{B}_v$ with $\text{momtsim}(P_i) = b \in \mathcal{B}$ the spectrum of feasible behaviors, and $p(b)$ the proportion of actors showing this behavior. Given that multiple sets of parameters can result in identical behavior, the production of a parameter set $\mathcal{P}(a_i)$ with $i \in \{1, \ldots, n\}$ for n agents means to choose among several parameter sets.

We select the parameter sets using the following criteria: $\mathcal{P}(a_i) = P_i \in \mathcal{P}_v$ with the probability $p(P_i)$ depending on the proportion of observed behaviors b and the number of parameter sets P_j leading to b. Hence,

$$p(P_i) = n \cdot \frac{p(b)}{|P_j|} \tag{13}$$

with

$$P_j \in \mathcal{P}_v | \text{momtsim}(P_i) = b. \tag{14}$$

Consequently, the probability of producing agent behaviors that were under-represented is increased because the likelihood of selecting a parameter set that is compatible with these behaviors is higher and the reverse holds for over-represented behaviors [12], [13]. Moreover, by utilizing $\mathcal{P}_v$ exclusively, without employing $\mathcal{P}$ all invalid parameter sets are removed.

3) PARAMETER SPACE EXPLORATION AND ITERATING THE METHOD

Parameter space exploration covers missing behaviors that might not otherwise be produced [12], [13]. Once missing behaviors have been established, we use an exploration function by choosing parameter sets in non-explored areas of the parameter set [12], [13]. This follows that

$$P(a_i) = P_i \in \mathcal{P}_v, \quad \text{if } p > \gamma \tag{15}$$

else

$$P(a_i) = P_k \notin \mathcal{P} \tag{16}$$

where γ is an exploratory parameter for searching new agent behaviors and p is a uniform random value. P_k is a candidate parameter set that is being evaluated to determine if it leads to a valid behavior. It ought not to have been selected in any of the previous steps of the parameter space exploration process. If P_k results in a valid agent behavior, it is included in the set of valid parameters $\mathcal{P}_v$ if not, it is discarded.

Once the desired behaviors have been reproduced in the simulations, iterating the calibration process potentially enables the discovery of new agent behaviors, fixing more errors using behavioral questionnaires on mobile money actors. A high-level view of the calibration process is shown in Fig. 3. The resulting synthetic datasets from the MoMTSim platform after successful calibration were evaluated against the real data to assess the closeness of their statistical qualities. Validation of the datasets enabled us to affirm the effective calibration of the numerous parameters of the simulation platform.

D. VALIDATION OF SYNTHETIC DATA

Validation of the synthetic data involved the computation of the difference between the real and synthetic data using the SSE method among other things. For various simulations, MoMTSim recorded varying total errors between the real and synthetic data, and the dataset with the least total error was chosen for analysis [19]. The aggregated transactions for the real and synthetic datasets were visualized to assess the closeness of the datasets.

1) VISUALIZATION OF ENTIRE MOBILE MONEY TRANSACTIONS

The distributions of the total transaction value, average transaction value, and the count of transactions were visualized for the real and synthetic data. The total transaction value and average transaction value were measured in virtual units (VUs) due to a non-disclosure agreement on the actual currency units. Violin plots were also used for visualization, combining box plots and density plots to compare the datasets. Violin plots are rich enough, showing both the box plot and the kernel density plot, unlike the use of only a box plot. Besides, it is easy to compare and contrast the distributions of real and synthetic data (including for large datasets), visually using violin plots [28], [29].

2) VISUALIZATION OF EACH MOBILE MONEY TRANSACTION TYPE

Data visualization was also performed for each transaction type (deposit, withdrawal, transfer, debit, and payment) for the total transaction value. This allowed for a detailed comparison of the real and synthetic data to draw insight into the validity of the simulation model.

IV. RESULTS

A. TOTAL ERRORS ACROSS DIFFERENT SIMULATIONS

Synthetic datasets arbitrarily named MoMTSim_202301, MoMTSim_202311, and MoMTSim_202312 with varying numbers of clients were generated. The MoMTSim_202311 simulation registered the least total error even though other simulations recorded low total errors as shown in Table 2. Nonetheless, the total errors registered by the different simulations are close to each other even though different numbers of clients were involved in the simulations. This indicates consistency in error propagation across different simulations of MoMTSim with varying counts of clients. MoMTSim_202311 was used to assess the resemblance of the total transaction value, average transaction value, and the count of transactions for the entire dataset to real data as well as examine the statistical similarities of the distinct transaction types.

TABLE 2. Total errors for different simulations with MoMTSim.

Simulation	Number of Clients	Total Error
MoMTSim_202301	1 000	2.8674
MoMTSim_202311	10 000	2.8073
MoMTSim_202312	100 000	3.4578

B. QUANTITATIVE VALIDATION OF SYNTHETIC DATA ACCURACY

In the evaluation of synthetic datasets meant to mimic real mobile money financial transactions, the KS test and Bland-Altman plots were used to assess the fidelity and agreement of the datasets.

Table 3 shows a comparison of the distribution of the transaction count, the total and average transaction values between the real and MoMTSim_202311 datasets. We report low KS statistics of 0.0213 for the transaction count and average transaction value, and 0.0142 for the total transaction value accompanied by high p-values of 0.838 for the transaction count and average transaction value, and 0.996 for the total transaction value. This indicates minor differences between the CDFs of the real and synthetic mobile money transaction datasets. These results suggest that the synthetic data closely mimics the distribution of the real data and confirms the effectiveness of the synthetic data generation process of MoMTSim in replicating these distributions.

Bland-Altman plots were constructed to further reveal the agreement between the real and synthetic datasets on a more granular, point-by-point basis as shown in Fig. 4. The majority of differences between the datasets for

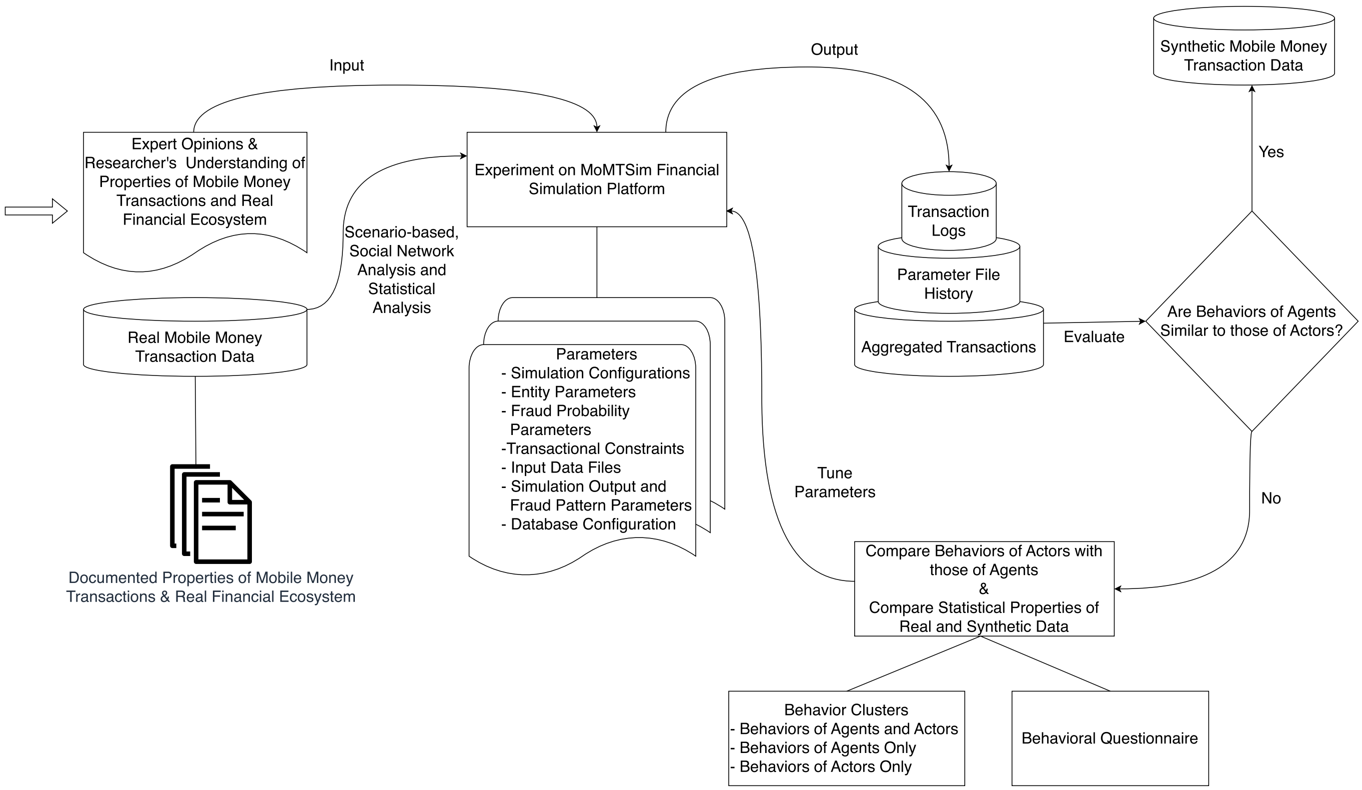


FIGURE 3. Calibration process for multi-agent-based simulation model for mobile money transactions.

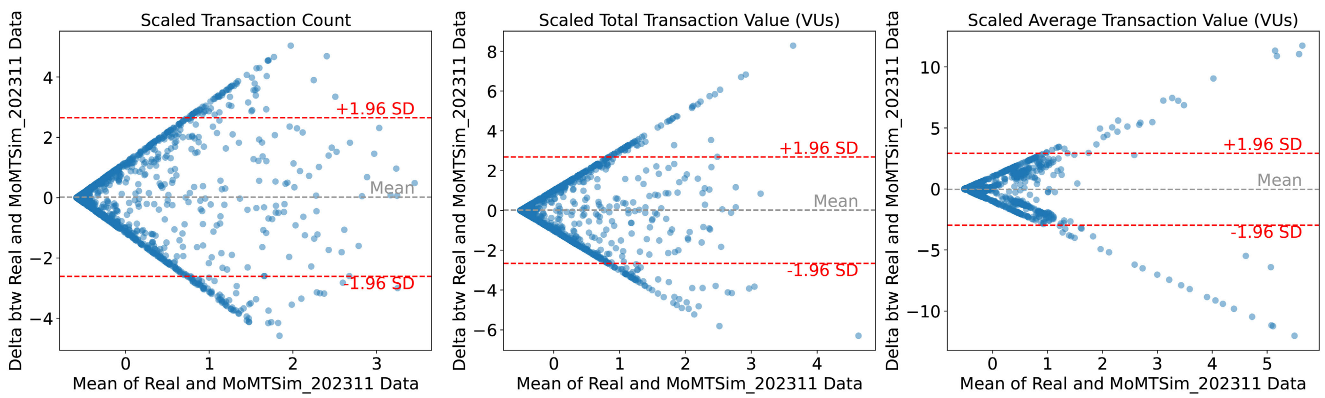


FIGURE 4. Bland-Altman plots for transaction count, total transaction value, and average transaction value between the real and MoMTSim_202311 data.

TABLE 3. KS statistics and p-values for transaction count, total transaction value, and average transaction value between the real and MoMTSim_202311 data.

Metric	KS Statistic	P-value
Transaction Count	0.0213	0.838
Total Transaction Value	0.0142	0.996
Average Transaction Value	0.0213	0.838

the transaction count, total transaction value, and average transaction value were consistently within the limits of the agreement. The clustering of data points around the mean difference line and the containment within the limits highlight a strong agreement between the datasets, with few exceptions. These findings affirm the validity of the MoMTSim model for the realistic synthesis of mobile money transaction datasets.

C. CONGRUENCE OF SYNTHETIC DATA TO REAL DATA

Fig. 5 shows the distributions of the total transaction value, average transaction value, and the transaction count for the entire real and synthetic data. The synthetic data exhibits a higher degree of overlap with the real data, both in terms of the extent of the overlapping areas and the similarity in their respective shapes with some differences in peak and tail densities. Minor differences in the shapes of the

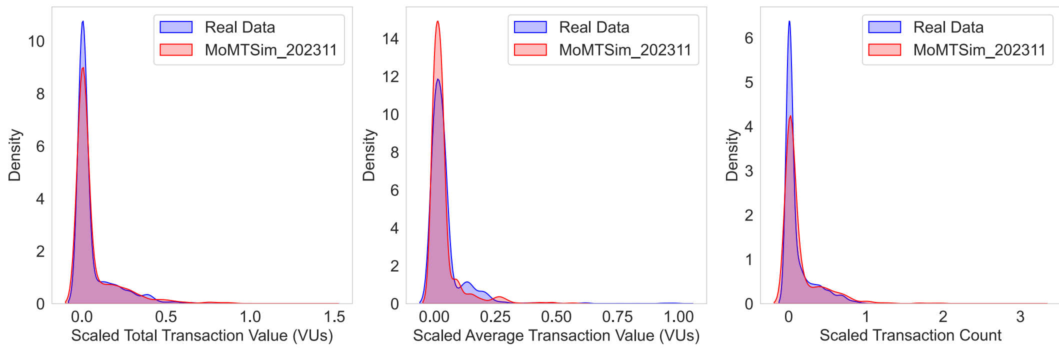


FIGURE 5. Distribution of real and MoMTSim_202311 data.

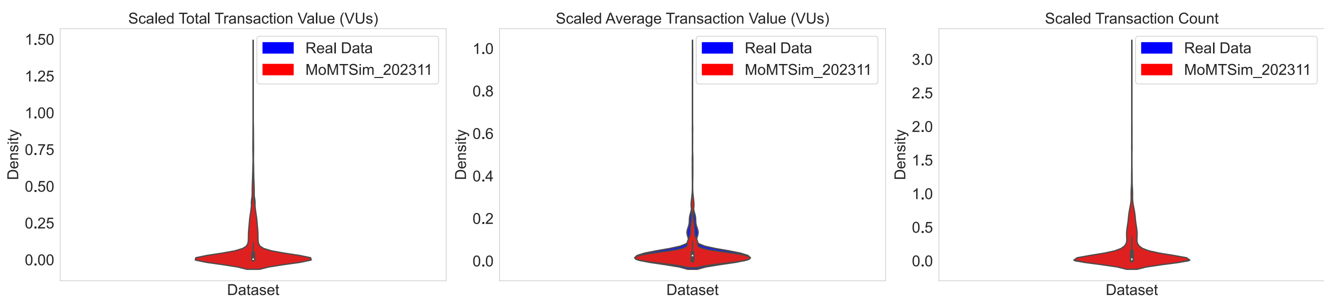


FIGURE 6. Violin plots for real and MoMTSim_202311 data.

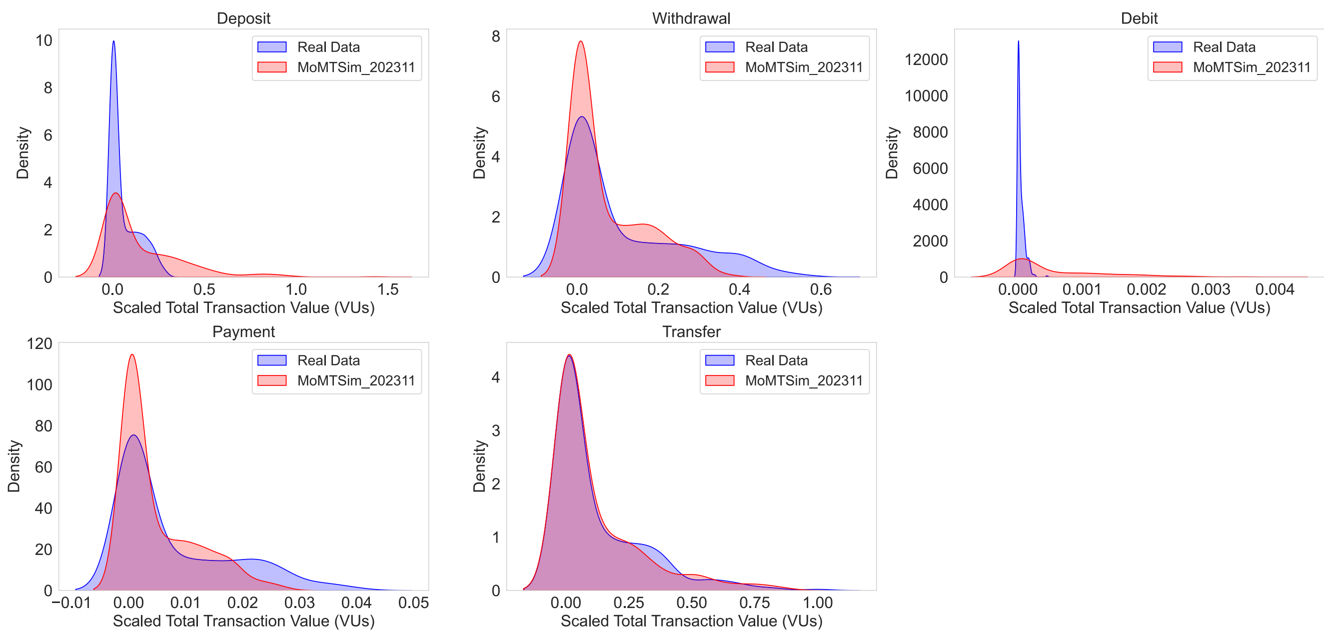


FIGURE 7. Distribution of total transaction values for the distinct transaction types in the real and MoMTSim_202311 data.

data are expected since synthetic data should resemble real data but not be a replication [30]. Slight variations in agent participation in carrying out transactions in MoMTSim to actor participation in the real financial ecosystem resulted in the minor differences presented. Fig. 6 shows the violin plots for the entire real and synthetic data. Similar shapes are observed indicating the resemblance of the datasets and this reinforces our earlier observations made from Fig. 5. To examine detailed patterns of each transaction type, we repeated the visualization process, for deposit, withdrawal, debit, payment, and transfer transactions as shown in Figs. 7 and 8. The degree of overlap in the

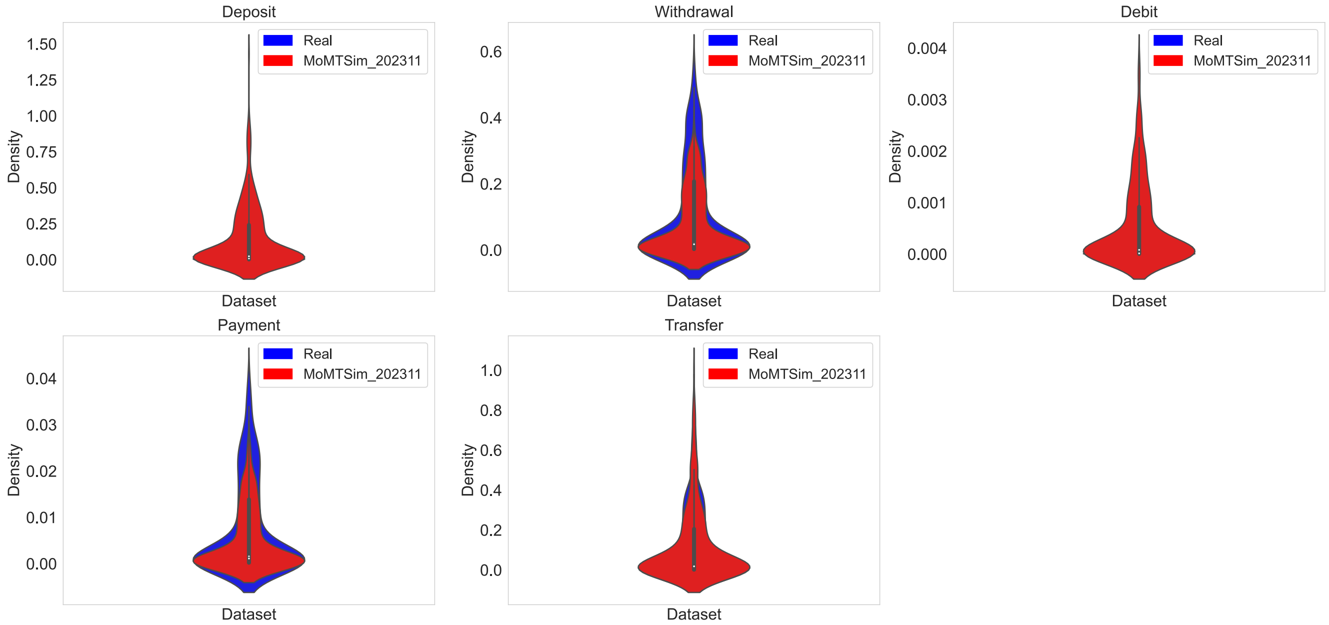


FIGURE 8. Violin plots for the total transaction values for the distinct transaction types in the real and MoMTSim_202311 data.

distributions of the different transaction types is significant indicating a high level of resemblance of the synthetic data to the real data. The differences in the tails and peak densities for the distinct transaction types imply the synthetic data is not a direct copy of the real data [30]. With an agent-based modeling approach, the synthetic data is a result of interactions between the agents, with varying behaviors. These differences prevent direct mapping of the synthetic data to real data [30]. The statistical similarities between the real and synthetic data confirm the validity of the simulation platform used for data synthesis. The synthetic data can be used to test hypotheses and train machine learning models for fraud detection especially when fraudulent activity has been introduced in the simulation platform.

Results in this study align closely with the findings and perspectives demonstrated in similar research [9], [10], [11], [12], [13], [30].

Our model calibration algorithm has been rigorously tested and validated within the context of a multi-agent mobile money financial system. However, its applicability to calibrate complex ecosystems beyond the financial domain remains unexplored and warrants further investigation. Additionally, potential computational costs associated with this algorithm were not examined in this study and could be addressed in future research. Despite these limitations, MoMTSim effectively produces substantial and realistic synthetic financial datasets, suitable for various financial analytics applications.

V. CONCLUSION

This study developed and demonstrated an iterative calibration algorithm for multi-agent systems, significantly improving the realism and accuracy of simulations such as the MoMTSim platform, which models mobile money transactions. Our method represents a departure from traditional static calibration techniques, allowing for dynamic adjustments based on real-world data and emergent behaviors. The model calibration approach relied on clusters of behaviors of agents and actors, behaviors of agents only, and behaviors of actors only. This was carried out with the help of a behavioral questionnaire and real transaction data for determining missing behaviors, removing invalid parameter sets, and adding new parameters in the agent model through the iterative calibration process.

This study successfully demonstrated the usage of the calibration algorithm in the MoMTSim platform for the synthesis of mobile money transactions. The degree to which the synthetic data generated with MoMTSim mirrors the real data was determined using the SSE method, KS test, and Bland-Altman plots. In particular, the SSE method computes the delta between the real and synthetic mobile money transaction datasets. We showed that the generated data statistically resembles the real data and this affirms our calibration approach. Also, the findings in this study are consistent with results from related studies [9], [10], [11], [12], [13], [30].

Future work could extend the validation of our model calibration algorithm to additional domains such as healthcare and urban planning. It will also be important to assess the computational costs associated with this method. Despite these limitations, our approach refines the accuracy of simulations, offering researchers and policymakers insights into complex systems and enabling the generation of extensive synthetic datasets for financial applications.

ACKNOWLEDGMENT

Any views or opinions expressed herein are solely those of the authors listed, and may differ from the views and opinions expressed by any funder or its affiliates. Several financial institutions in the Sub-Saharan region provided expert opinions on the dynamics of the real mobile money financial ecosystem.

REFERENCES

[1] M. Singh, A. Marathe, M. V. Marathe, and S. Swarup, "Behavior model calibration for epidemic simulations," in *Proc. Int. Joint Conf. Auto. Agents Multiagent Syst., AAMAS. Int. Joint Conf. Auto. Agents Multiagent Syst.*, 2018, p. 1640.

[2] N. Groves-Kirkby, E. Wakeman, S. Patel, R. Hinch, T. Poot, J. Pearson, L. Tang, E. Kendall, M. Tang, K. Moore, S. Stevenson, B. Mathias, I. Feige, S. Nakach, L. Stevenson, P. O'Dwyer, W. Probert, J. Panovska-Griffiths, and C. Fraser, "Large-scale calibration and simulation of COVID-19 epidemiologic scenarios to support healthcare planning," *Epidemics*, vol. 42, Mar. 2023, Art. no. 100662.

[3] A. Züfle, C. Wenk, D. Pfoser, A. Crooks, J.-S. Kim, H. Kavak, U. Manzoor, and H. Jin, "Urban life: A model of people and places," *Comput. Math. Org. Theory*, vol. 29, no. 1, pp. 20–51, 2023.

[4] S. F. Moosavi, N. Salehnia, A. Seifi, A. AsgharpourMasouleh, and N. Salehnia, "Designing and calibrating an agent-based platform to evaluate the effect of climate variables on residential water demand," *Water Environ. J.*, vol. 37, no. 3, pp. 604–615, Aug. 2023.

[5] K. M. Ullah, G. A. Oladosu, and A. Crooks, "Evaluating the incentive for soil organic carbon sequestration from carinata production in the Southeast United States," *J. Environ. Manage.*, vol. 348, Dec. 2023, Art. no. 119418.

[6] B. Shapiro and A. Crooks, "Drone strikes and radicalization: An exploration utilizing agent-based modeling and data applied to Pakistan," *Comput. Math. Org. Theory*, vol. 29, no. 3, pp. 415–433, Sep. 2023.

[7] F. Shoushtarian, M. Negahban-Azar, and A. Crooks, "Investigating the micro-level dynamics of water reuse adoption by farmers and the impacts on local water resources using an agent-based model," *Socio-Environ. Syst. Model.*, vol. 4, p. 18148, Oct. 2022.

[8] N. Jiang, A. T. Crooks, H. Kavak, A. Burger, and W. G. Kennedy, "A method to create a synthetic population with social networks for geographically-explicit agent-based models," *Comput. Urban Sci.*, vol. 2, no. 1, p. 7, Feb. 2022.

[9] E. A. L. Rojas, S. Axelsson, and D. Baca, "Analysis of fraud controls using the PaySim financial simulator," *Int. J. Simul. Process Model.*, vol. 13, no. 4, p. 377, 2018.

[10] E. A. Lopez-Rojas and C. Barneaud, "Advantages of the PaySim simulator for improving financial fraud controls," in *Proc. Intell. Comput. Comput. Conf.*, vol. 2. Cham, Switzerland: Springer, 2019, pp. 727–736.

[11] E. Altman, "Synthesizing credit card transactions," in *Proc. 2nd ACM Int. Conf. AI Finance*, Nov. 2021, pp. 1–9.

[12] K. Darty, J. Saunier, and N. Sabouret, "Agents behavior semi-automatic analysis through their comparison to human behavior clustering," in *Proc. Int. Conf. Intell. Virtual Agents*, Boston, MA, USA. Cham, Switzerland: Springer, Aug. 2014, pp. 154–163.

[13] K. Darty, J. Saunier, and N. Sabouret, "Calibration of multi-agent simulations through a participatory experiment," in *Proc. 14th Int. Conf. Auto. Agents Multiagent Syst.*, 2015, pp. 1683–1684.

[14] H. Van Dyke Parunak, R. Savit, and R. L. Riolo, "Agent-based modeling vs. equation-based modeling: A case study and users' guide," in *Proc. Int. Workshop Multi-Agent Syst. Agent-Based Simulation*. Cham, Switzerland: Springer, 1998, pp. 10–25.

[15] P. O. C. Xavier, A. P. F. Atman, and A. R. B. de Magalhães, "Equation-based model for the stock market," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 96, no. 3, Sep. 2017, Art. no. 032305.

[16] S. Daun, J. Rubin, Y. Vodovotz, and G. Clermont, "Equation-based models of dynamic biological systems," *J. Crit. Care*, vol. 23, no. 4, pp. 585–594, Dec. 2008.

[17] *MoMTSim Financial Simulation Platform*, document version 1.0.0, Dept. Comput. Sci., Makerere Univ., 2023. [Online]. Available: https://github.com/aiinfinancegroup/MoMTSim

[18] K. Sullivan, M. Coletti, and S. Luke, "GeoMason: Geospatial support for mason," Dept. Comput. Sci., George Mason Univ., Fairfax, VA, USA, Tech. Rep. GMU-CS-TR-2010-16, 2010. [Online]. Available: https://cs.gmu.edu/

[19] E. A. Lopez-Rojas, A. Elmir, and S. Axelsson, "PaySim: A financial mobile money simulator for fraud detection," in *Proc. 28th Eur. Modeling Simulation Symp. (EMSS)*, Sep. 2016, pp. 249–255.

[20] C.-V. Pal, F. Leon, M. Paprzycki, and M. Ganzha, "A review of platforms for the development of agent systems," 2020, *arXiv:2007.08961*.

[21] S. Luke, C. Cioffi-Revilla, L. Panait, and K. Sullivan, "MASON: A new multi-agent simulation toolkit," in *Proc. Swarmfest Workshop*, vol. 8, no. 2. Lansing, MI, USA, 2004, pp. 316–327.

[22] S. Tisue and U. Wilensky, "NetLogo: A simple environment for modeling complexity," in *Proc. Int. Conf. Complex Syst.*, vol. 21. Princeton, NJ, USA: Citeseer, 2004, pp. 16–21.

[23] N. Collier, "RePast: An extensible framework for agent simulation," in *University of Chicago's Social Science Research*, vol. 36. Chicago, IL, USA: The Univ. of Chicago, Jun. 2003, p. 2003.

[24] Y. G. Karpov, "AnyLogic: A new generation professional simulation tool," in *VI Int. Congr. Math. Model., Nizni-Novgorog, Russia*, 2004.

[25] M. J. Hayat and M. Higgins, "Understanding Poisson regression," *J. Nursing Educ.*, vol. 53, no. 4, pp. 207–215, Apr. 2014.

[26] H.-H. Bock, "Origins and extensions of the k-means algorithm in cluster analysis," *Electron. J. Hist. Probab. Statist.*, vol. 4, no. 2, pp. 1–18, 2008.

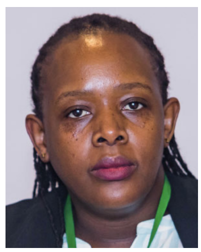
[27] D. Du, *Social Network Analysis: Centrality Measures*. Fredericton, NB, Canada: Univ. New Brunswick, 2019.

[28] R. Tanious and R. Manolov, "Violin plots as visual tools in the meta-analysis of single-case experimental designs," *Methodology*, vol. 18, no. 3, pp. 221–238, Sep. 2022.

[29] D. T. Aberasturi, "Violin plot," in *Wiley StatsRef: Statistics Reference Online*. Hoboken, NJ, USA: Wiley, 2014, pp. 1–7.

[30] S. A. Assefa, D. Dervovic, M. Mahfouz, R. E. Tillman, P. Reddy, and M. Veloso, "Generating synthetic data in finance: Opportunities, challenges and pitfalls," in *Proc. 1st ACM Int. Conf. AI Finance*, New York, NY, USA, Oct. 2020, pp. 1–8.

DENISH AZAMUKE received the B.Sc. degree in computer science from Makerere University, Uganda, in 2020, and the M.Sc. degree in mathematical sciences from African Institute for Mathematical Sciences, Rwanda, in 2021. He is currently pursuing the Ph.D. degree in computer science, under the supervision of Dr. Marriette Katarahweire and Dr. Engineer Bainomugisha. Since 2022, he has been a Graduate Fellow and a Researcher with the Department of Computer Science, Makerere University. He is also a Google Ph.D. Fellow at Makerere University. His research interests include synthetic data generation, privacy-preserving technologies, machine learning, and artificial intelligence.



MARRIETTE KATARAHWEIRE is currently a Postdoctoral Researcher and a Lecturer with the Department of Computer Science, School of Computing and Informatics Technology, College of Computing and Information Sciences, Makerere University, Kampala, Uganda. Her research interests include security and privacy-enhancing technologies for synthetic data generation in finance and mobile health.

ENGINEER BAINOMUGISHA is currently a Faculty Member and the Chair of the Department of Computer Science, Makerere University. His research interests include distributed systems, programming languages, the Internet of Things, cloud computing, machine learning, and artificial intelligence for social good.