Proceedings of the 2024 Winter Simulation Conference
H. Lam, E. Azar, D. Batur, S. Gao, W. Xie, S. R. Hunter, and M. D. Rossetti, eds.

A FRAMEWORK FOR DIGITAL TWIN COLLABORATION

Zhengchang Hua[1,2], Karim Djemame[2], Nikos Tziritas [3], and Georgios Theodoropoulos[4,1]

[1]Dept. of Computer Science and Eng., Southern University of Science and Technology (SUSTech), Shenzhen, CHINA
[2]School of Computing, University of Leeds, Leeds, UK
[3]Dept. of Informatics and Telecommunications, University of Thessaly, Lamia, GREECE
[4]Research Institute of Trustworthy Autonomous Systems (RITAS), SUSTech, Shenzhen, CHINA

ABSTRACT

Digital Twins (DTs) have emerged as a powerful tool for modeling Large Complex Systems (LCSs). Their strength lies in the detailed virtual models that enable accurate predictions, presenting challenges in traditionally centralized approaches due to the immense scale and decentralized ownership of LCSs. This paper proposes a framework that leverages the prevalence of individual DTs within LCSs. By facilitating the exchange of decisions and predictions, this framework fosters collaboration among autonomous DTs, enhancing performance. Additionally, a trust-based mechanism is introduced to improve system robustness against poor decision-making within the collaborative network. The framework's effectiveness is demonstrated in a virtual power plant (VPP) scenario. The evaluation results confirm the system's objectives across various test cases and show scalability for large deployments.

1 INTRODUCTION

The modeling and governance of Large Complex Systems (LCSs) have received significant attention in recent years. Examples of LCSs include power grids, smart cities, regional networks, and even the Earth itself. By definition, LCSs are intricate systems composed of a large number of interacting parts (Waldrop 1993). Due to the immense scale and intricate interactions of their member entities, the overall behavior of LCSs cannot be easily predicted solely by understanding the individual parts.

Digital Twins (DTs) have garnered significant attention as a promising approach for modeling LCS in different domains (Ghandar et al. 2021; Diamantopoulos et al. 2024; Diamantopoulos et al. 2022). They are not merely digital representations of physical assets, but rather "live" models that are continuously updated with real-time data reflecting the dynamic changes of their physical counterparts (Grieves and Vickers 2017). Beyond mirroring physical entities, DTs enable the simulation of potential outcomes for various actions applied to the physical system, facilitating "what-if" analysis. Through this process, DTs can not only identify the optimal operation for a given objective but also assess the potential consequences of human interventions (Diamantopoulos et al. 2023; Zhang et al. 2024b; Zhang et al. 2020).

Constructing a single DT for an LCS presents several challenges. Firstly, due to the simulation-centric nature of DTs, a highly detailed model of the entire LCS is necessary to achieve an accurate digital representation. However, such large models create significant pressure on real-time data integration, simulation speed, and the ability to perform real-time "what-if" analysis. While distributed and large-scale simulation (such as HLA) may address scalability by partitioning the model and corresponding data across multiple processors (Logan and Theodoropoulos 2000; Lees et al. 2005; Ventresque et al. 2012; Suryanarayanan and Theodoropoulos 2013), it may not be always desirable to have such closely synchronized models of systems at so many different spatiotemporal scales.

Another important factor that would hinder the utilization of such simulation approaches is privacy and security. In a typical LCS such as a smart city, there are many different entities owned by different stakeholders. For example, power supplies in different regions are often controlled by different stakeholders, and these stakeholders may be hesitant to share information with other parties due to privacy and security concerns (Ismagilova et al. 2022). Federated learning may be a potential solution to the privacy challenge allowing individual members to train models separately using their own data (McMahan et al. 2017). The trained models from all members are then aggregated to create a larger model that achieves similar effectiveness to a centrally trained model. However, a key limitation of federated learning in the context of DTs is that many machine learning models remain opaque, hindering their use for "what-if" analysis and limiting explainability compared to traditional simulation techniques (Zhang et al. 2024a).

To cope with these challenges various approaches have increasingly been proposed to enable collaboration of multiple independent DTs. In (Vergara et al. 2023) the authors introduced the concept of Federated Twins and presented a general framework to enable DT federations. This paper proposes an approach to enable collaboration among multiple DTs in the context of such federations. By exchanging only high-level information like predictions and decisions, raw data sharing is not required for DTs in the collaboration. Due to the reduced scale of information exchange, the framework does not require complex data or time synchronization between DTs, and the coupling between individual DTs is reduced. As different DTs use different models and may have different perspectives on the physical world, it is not uncommon that DTs generate inaccurate predictions on certain physical areas or certain properties. To enhance system stability and mitigate the impact of inaccurate predictions, a trust mechanism is introduced that considers the alignment between DT predictions and real-world outcomes. The proposed framework may be implemented as a generic middleware that runs on top of the existing DT instances, minimizing the modifications required for them to participate in the collaborative environment.

The main contributions of this paper are as follows:

- A novel framework is proposed that facilitates collaboration among DT instances within a LCS. This framework enables the sharing of predictions and decisions between DT instances while preserving the privacy of their data and maintaining their independence.
- A trust-based mechanism is introduced for mitigating the impact of inaccurate predictions thus enhancing the robustness of decisions.
- The proposed framework and mechanism are demonstrated and analyzed in the context of a DT federation of a smart grid.

The remainder of the paper is organized as follows. Section 2 provides an overview of related work. Section 3 presented the proposed approach, detailing the system model, architecture design, and collaboration procedure, and introducing the proposed trust mechanism to reduce the impact of inaccurate predictions and decisions made by DTs, thus improving the robustness of the system. Section 4 evaluates the effectiveness of the proposed framework presenting quantitative results in the context of a smart grid case study. Finally, Section 5 concludes the paper and outlines future work.

2 RELATED WORK

One key technology used to facilitate the collaboration between DTs is federated learning. In the work by San et al. (2023), a decentralized DT modeling work is proposed based on the idea of federated learning. In their work, they investigated the training of physics-guided neural networks and nonlinear dimensionality reduction of dynamical systems using federated learning, and demonstrated the effectiveness of federated learning in complex nonlinear spatiotemporal systems. Pang et al. (2021) proposed a framework that applies federated learning to city DTs to enable the sharing of local strategy and status. In this way, a collaborative city DT could obtain knowledge from multiple DTs and establish a global view of city management, as well as improve individual DTs locally. The use of federated learning technology has tackled the problem of data privacy and security, and has shown to be effective in training models in a decentralized manner. However, compared to simulation, federated learning lacks two critical features that are essential for DTs:

what-if analysis and explainability of decisions. Thus, to construct an explainable and what-if analysis enabled collaborative DTs system, the issues of data privacy and security on the use of simulation need also be resolved.

Several studies explore the use of blockchain technology to facilitate communication between DTs. Sahal et al. (2022) propose a blockchain-based framework for collaborative DTs in decentralized epidemic alerting. Their work enables DTs to share historical data and collaborate on predicting epidemic outbreaks in a decentralized manner using blockchain. Another relevant framework, CoTwin (García-Valls and Chirivella-Ciruelos 2024), leverages blockchain to create a secure and immutable storage space for training DT machine learning models. The distributed global ledger offered by blockchain allows DTs to trace the characteristics of their machine learning models. While blockchain demonstrably offers secure and trustworthy universal data exchange and sharing, it presents challenges regarding data privacy. Participating DTs may be hesitant to share raw private data, even if the sharing process itself is secure. Additionally, in large, complex systems that generate massive amounts of data, utilizing a distributed ledger could introduce significant storage overhead and potentially hinder the real-time execution of DTs.

Another approach to building a collaborative DT is to organize the DTs in a hierarchical manner. Villalonga et al. (2020) designed and implemented a distributed DT framework to improve decision making at the local level in manufacturing processes. Their system consists of several local DTs to do local decision making, and all local DTs are linked to a global DT, which does the scheduling and global decision making. Lombardo and Ricci (2022) introduced a federation of DTs as a platform to visualize the disease of a patient as a whole. They established a DT model for each organ in a human, and on top of the individual DTs is a Second Level DT which has a global vision on all the DTs. It keeps a copy of data about every single DTs connected to it, and uses AI to monitor the functioning of the entire system of DTs. This approach utilizes the existing infrastructure of DTs, and provides a way to organize the DTs hierarchically. However, the hierarchical structure may not be suitable for all applications, and the communication between DTs at different levels may be difficult to manage. Also, raw data from DT instances needs to be shared with the global DT, which may raise concerns about data privacy and security.

Work that uses multi-agent systems to facilitate the collaboration between DTs is also worth mentioning. Hui et al. (2022) developed a DT-enabled architecture to help autonomous vehicles make driving decisions in collaborative driving scenarios. They proposed two mechanisms to assist collaborative driving, one deciding the role of DTs in DT groups, and another deciding the optimal group distribution. Moshrefzadeh et al. (2020) introduced a concept called distributed DT in the agricultural landscape, which handles resources from different stakeholders, and provides a basis for integrating information from physical assets, including pre-existing, historical, and real-time data. They developed a metadata model for the distributed DT, to enhance the applicability and usability of the relationships between different resources from different stakeholders. This approach treats DTs as agents, and leverages multi-agent systems to facilitate the collaboration between DTs. However, this limits the autonomy of individual DTs, and may not be suitable for applications where DTs need to make decisions independently. It also reduces the robustness of the system, as the failure of one DT may affect the performance of the whole system.

While current technologies provide a solid foundation for effective DT collaboration, their application in LCSs involving multiple DTs with different ownerships remains limited. In contrast, the approach proposed in this paper facilitates the collaboration of multiple DTs by sharing information at the level of predictions and decisions. This method reduces the data that needs to be exchanged between DTs, allows them to make independent local decisions while contributing to global objectives, and helps protect the private information of each DT.

3 A FRAMEWORK FOR DIGITAL TWIN COLLABORATION

The proposed collaboration framework is designed as a distributed middleware that can seamlessly integrate with existing DTs, enabling their collaboration. The system supports fully decentralized DTs. Each DT interacts with its own framework middleware, while middleware instances interact using broadcasting and multicasting, fostering scalability and robustness. The framework can operate as a plugin, adaptable to existing DTs with minimal modifications to their internal structure.

The main components of the middleware are shown in Figure 1. The **Communicator** acts as the conduit for information exchange between DTs within the collaborative network. It ensures a smooth flow of communication, enabling them to share predictions and decisions effectively. The **DT Interface** addresses the challenge of diverse communication protocols. It functions as a translator, adapting domain-specific DTs to the framework's common language, ensuring clear and consistent understanding. The **History Database** holds the collective memory of the system and stores historical data on predictions and decisions made by individual DTs. This historical knowledge is the key to the trust mechanism, enabling DTs to evaluate the reliability of their peers. The **Verifier** safeguards accuracy by meticulously comparing predictions from each DT with real-world data. This rigorous verification process identifies any inconsistencies or potential issues, upholding the reliability of the overall system. The **Scope Manager** helps to manage the communication scope of the communicator to reduce unnecessary messages to be sent to the DTs that are not in the scope of influence of some predictions or decisions.

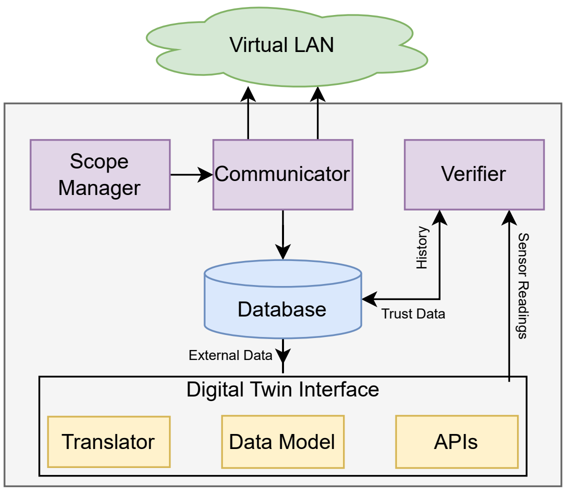

Figure 1: Middleware components.

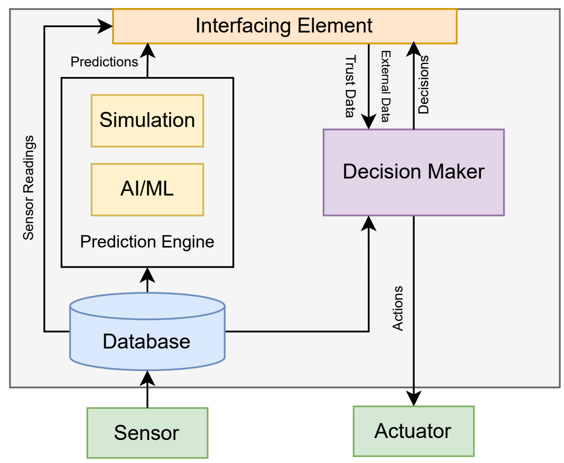

Figure 2: Reference DT components.

While the framework design doesn't include the inner components of DTs, the underlying DT of the middleware needs to have some essential components. Figure 2 shows a reference model of the DT used by the framework. **Sensor** and **Actuator** is the interface that DT interacts with the physical world. **Database** stores the sensor readings gathered by the sensors, and provides data to the prediction engine. Predictions are generated by the **Prediction Engine**, which typically is a simulator, but could also include machine learning tools. The **Decision Maker** is responsible for making decisions based on the predictions and other information. It actively evaluates the situation based on the data from sensors or the middleware, and also updates its own decisions to actuators to implement or to middleware to share with other DTs. The **Interfacing Element** bridges the DT to the middleware.

3.1 Communication

Effective collaboration among DTs hinges on a communication approach that facilitates seamless information exchange. However, achieving this can be challenging due to the inherent domain-specificity and diverse operational scenarios of these twins. To overcome this hurdle, a common communication ground is essential. This framework introduces the Communicator component, which serves as a standardized interface for DT interaction and information exchange. This component provides a unifying layer, regardless of the specific domain or application context. To establish a secure and manageable network infrastructure, the framework leverages virtualized network technology. In this way, all participating DTs are placed within the same virtual network, eliminating the complexities associated with traditional naming and registry services. To ensure a unified communication protocol, the framework defines the following key components:

Base Model. The base model forms the common basis for communication between DTs. It defines a way to represent real-world entities through a list of states. Each state is a tuple consisting of three elements: a globally unique identifier for clear identification, a human-readable name for easy understanding, and the actual value reflecting the current state. While states often have relationships based on their physical or logical connections, the base model does not define these relations. Instead, the relationships are modeled internally within each DT.

In a system of multiple DTs, each DT possesses internal attributes that may be security-sensitive or irrelevant to other DTs, residing solely within that specific instance. These states are called **internal states**. Conversely, some states are visible to other DTs, often located at subsystem boundaries, and play a role in overall system behavior by allowing multiple DTs to monitor and influence them. These states are defined as **interfacing states**. An example of interfacing states is the electricity being transmitted between two nodes through a power line. Both DTs of the entities can take measurements of the electricity flow through sensors, and the flow can be adjusted by any of the DTs by sending commands to its actuators related to voltage adjustment.

Since the list of states is closely related to the physical or logical entities in the real world, in most cases they will not be added or removed frequently. Therefore, we assume that the list of states is pre-defined and agreed upon by all DTs offline, and the list of states is not changed during the runtime.

Prediction. Each prediction corresponds to a predicted future value of a state. The prediction is represented as a tuple, each consists of three values: time of the prediction, the globally unique identifier of the state, and the value of the state. To reduce the communication load, predictions that belong to the same time frame or belong to the same state with different timestamps can be grouped as a set, thus avoiding sending duplicate information.

Decision. For a decision, we define it as an action and the related predictions after the action has been performed. The action, which is the change to a specific state at a future time, has the same format as the prediction but is identified with different message headers. Actions and the related predictions are packed and sent together as a decision, and the same as the prediction, the decisions could be grouped as a set to reduce the communication load.

3.2 Interaction

When operational, the framework facilitates a continuous exchange of information. Each DT periodically broadcasts its predictions to collaborators within the federation. These predictions are received by participating DTs, whose decision-making modules then incorporate this information to refine their own local predictions. This collaboration offers two key advantages: improved local predictions through the use of updated interfacing states based on collaborator predictions, and proactive action planning enabled by high-confidence predictions about future states derived from decisions made by other DTs. Following suit, upon reaching a new local decision, each DT broadcasts both the decision itself and the associated predictions to the network. Collaborators then receive and utilize this information to update their own decision-making processes, perpetuating this cyclical exchange.

Here, it's crucial to acknowledge the concept of locality. In many conditions, DTs do not require broadcasting predictions and decisions to every member of the system. By leveraging the mapping between global states and specific instances, DTs can identify relevant collaborators based on their predictions and decisions. Consequently, broadcasts are targeted only to these relevant collaborators, reducing communication overhead and enhancing the system's scalability.

3.3 Trust Mechanism

While sharing predictions and decisions fosters knowledge exchange among collaborating DTs, it also introduces potential vulnerabilities. Even in trusted environments, free from malicious actors, errors in predictions can still occur. To mitigate this risk, the framework incorporates a trust mechanism that assesses the accuracy of shared predictions from collaborators and establishes a credit history for each DT instance. This information empowers decision-makers to weigh information from other instances more effectively.

The process of verifying predictions and calculating trust score is shown in Algorithm 1. The entire process includes four major steps: mapping prediction value to reality sensor readings, aggregating the mapped historical values by time frames, calculating the similarity score of each time frame and finally the trust score. The verification and trust calculation procedure is handled by a dedicated verifier component in the middleware. The verification process begins with the creation of a mapping between predictions and real-world data. Periodically, the verifier reevaluates past predictions using newly available sensor data. The algorithm iterates through all sensor readings (line 6) and previously recorded predictions for each state of each DT. It then compares the timestamps of these readings and predictions to identify those with a temporal difference that falls within the predefined threshold (line 8). The threshold defines the maximum allowable time difference between a sensor reading and a prediction for them to be considered temporally relevant for comparison. Its value depends on the specific application and aligns with the DT's simulation timestep. If the check passes, it will establish a mapping between the selected prediction and observation, and put them into the mapping set. It repeats this procedure until all possible mappings are made. Next, to capture a snapshot of a DT in a time frame, the verifier will try to group all states of the same DT by time frames (line 15~22). After this aggregation process, the verifier will generate a list of time frames, with each time frame consisting of all mappings of predictions and observations over all the states of a DT.

After obtaining the time-sliced view of each DT, prediction accuracy will be calculated using a similarity function (line 25), which calculates Euclidean distance between the real measurement and predictions, and normalize the results using the Gaussian kernel function, as shown in Equation 1. In the equation, we use the same notation as in Algorithm 1. h_p and h_r represent the prediction and observation, respectively. F_i is the time frame generated in Step 2 of the algorithm. Parameter σ is the width of the Gaussian kernel which is determined by the specific application scenario. The resulting similarity score, along with the corresponding timestamp, is then stored in the history database for later use, or to be used immediately to calculate the trust score.

$$sim_i = \exp\left(-\frac{\sum_{(h_p,h_r)\in F_i}(h_p - h_r)^2}{2\sigma^2}\right) \tag{1}$$

Leveraging this historical data on similarity scores, the trust mechanism calculates a trust score for each DT instance (line 31), which is shown in Equation 3. The trust calculation incorporates both the similarity score and the time elapsed since the score was calculated. The trust score is a weighted average of similarity scores over a past period, and the calculation of weight is shown in Equation 2. Older similarity scores contribute less weight to the overall trust value, with the parameter α governing the rate at which this influence diminishes. The decision-making module then utilizes these trust scores to weigh the influence of predictions received from different collaborators.

$$w_i = \frac{1}{1+\alpha(t_{now} - t_i)} \tag{2}$$

$$trust = \frac{\sum_{(t_i,sim_i)\in sim} sim_i \cdot w_i}{|sim|} \tag{3}$$

Algorithm 1 Verification and Trust score calculation.

Input: d: target DT, H_P: set of historical prediction data, H_R: set of historical sensor data, S_d: set of relevant states of d
Initialization: trust score $t = 0$, prediction-reality mapping $M = \varnothing$, time frame mapping $F = \varnothing$, similarity score set $sim = \varnothing$
Step 1: **Prediction-Reality Mapping**
for each $s \in S_d$
  $M_s = \varnothing$
  for each $(h_r, h_p) \in H_R \times H_P$
    extract timestamp t_{h_r} *and* t_{h_p} *from* h_r, h_p
    if $|t_{h_p} - t_{h_r}| \leq t_{thres}$ **then**
      $M_s = M_s \cup \{(t_{h_r}, h_p, h_r)\}$
      break
    end if
  end for
end for
Step 2: **Time Frame Mapping**
split timespan to discrete time frames
for each time frame i
  $F_i = \varnothing$
  for each M_s in M
    Add $\{(t_{h_r}, h_p, h_r)\}$ *to* M_i *if* t_{h_r} *is in time frame* i
  end for
  $F = F \cup \{F_i\}$
end for
Step 3: **Similarity Score Calculation**
for $F_i \in F$
  sim_i =*calculateSimilarity*(F_i)
  calculate the time t_i *between* F_i *and now*
  $sim = sim \cup \{(t_i, sim_i)\}$
end for
sort sim by t in descending order
Step 4: **Trust Score Calculation**
calculateTrustScore(sim)

4 EVALUATION

This section provides a quantitative evaluation of the proposed framework in terms of performance and communication overhead. A federation of DTs for the management of a Smart Grid is used as a use case.

4.1 Experiment Setup

A smart grid is a modernized power grid that leverages information and communication technology to gather and utilize data, such as supplier and consumer behavior patterns, in an automated fashion (Fang et al. 2011). This automation aims to enhance the efficiency, reliability, economics, and sustainability of electricity production and distribution. One promising technology for smart grids is the Virtual Power Plant (VPP). A VPP functions as a distributed power plant by aggregating the capacities of diverse Distributed Energy Resources (DERs) to augment overall power generation (Pudjianto et al. 2007). To showcase the effectiveness of our proposed framework, a VPP has been developed that aggregates DER capacities to address the power demands of a specific community.

Our experimental setup models each DER as a DT within a collaborative federation that constitutes the VPP. Four DER types are incorporated: solar panels, wind turbines, hydropower turbines, and diesel generators. Except diesel generators, each renewable energy source is equipped with a battery unit for temporary energy storage. The power output of solar panels and wind turbines is inherently variable, depending on weather conditions like wind speed and solar irradiance. Similarly, hydropower turbine output is dictated by the water flow rate. While direct control over these outputs may be limited, the storage units enable short-term adjustments within a narrow range. In contrast, diesel generators offer controllable power output through adjustments to fuel input. The VPP's objective within this framework is to consistently satisfy the community's power demand under various conditions.

For the experiment testbed, we implemented minimal, in-house DTs that operate entirely within a simulated environment. All sensor and actuator data for these DTs originates from the simulation itself. Environmental factors, like weather, are modeled using historical data (traces), while actuator actions are reflected within the simulation and subsequently detectable by sensors. The decision-maker within each DT employs a simple load-following policy. For diesel generators, this policy adjusts power output dynamically based on the difference between current demand and available power supply, with a limit on the speed of adjusting the output power. For other DERs, the policy dictates when and how to utilize battery storage units. This can involve storing excess energy or drawing from the batteries to compensate for fluctuations in power output. For simplicity, the detailed energy flow between DERs and consumers is not modeled. All DERs in the system output power to a main bus and each consumer draws power from the bus. It is also worth noting that the system setup allows excess power to be generated. This is based on the fact that VPPs can usually sell their excess power to larger grids.

To comprehensively evaluate our framework's effectiveness, we establish three distinct small-scale scenarios and one large-scale scenario to assess scalability. The first scenario simulates a power ramp-up event, where the community's power demand abruptly increases at the simulation's outset. The VPP must then elevate its power output to meet the new demand. The second scenario explores the VPP's response to failures. Here, we simulate the concurrent malfunction of two wind turbine DERs due to strong winds. The VPP must compensate for this unexpected loss of generation capacity and maintain sufficient power output to meet demand. The third scenario investigates the VPP's ability to detect and address inaccurate predictions. We simulate a situation where one wind turbine DER generates erroneous power predictions. The remaining DERs must recognize this anomaly and adjust their output accordingly to ensure the community's power needs are met. In the first two scenarios, we compare the baseline, where DTs act purely based on local goals, and the collaborative case, where DTs exchange information using the framework. The third scenario compares the framework without the trust mechanism to the framework with the trust mechanism enabled. Finally, the scalability test investigates the framework's overhead as the number of DERs increases. This test aims to verify the framework's capability to handle growing system complexity. The experiment is run on a laptop with an AMD Ryzen™ 7 7840HS CPU and 32GB of RAM. Each set of parameters is tested 10 times.

4.2 Result analysis

In Figure 3 and Figure 4, we compare the baseline case where each DT acts individually (*Base*), and our collaborative DT (*CDT*) as described in Section 3.2 but without trust mechanism. For clarity, the numerical value of the power output has been normalized, with 1.0 as the target value. Figure 3 shows the power ramp-up process of Base and CDT. From the figure, we can see that CDT has a higher ramp-up speed compared to the baseline, taking 105 seconds to ramp up to the target power output, whereas the baseline takes 175 seconds to reach the new goal. The performance improvement is primarily due to the communication with the diesel fuel generator, which allows other DTs to be aware of the ramp-up speed and the expected time for the diesel fuel generator to reach the target power output. They could, therefore, take a more aggressive approach, using existing energy storage units to ramp up the output more quickly.

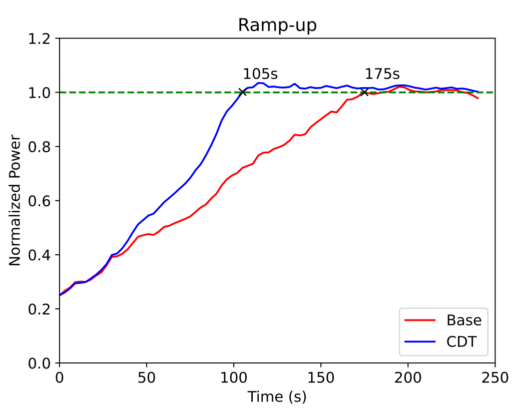

Figure 3: The ramp-up scenario.

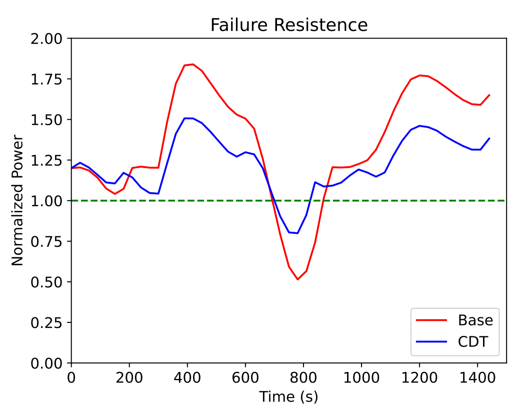

Figure 4: The failure scenario.

Figure 4 demonstrates how failure forecasting and handling are improved by the integration of the collaborative mechanism. In the scenario setup, a subsequent failure of two wind turbine DERs because of overspeed is modeled. The simulation is run for 24 minutes, and the Y axis shows the maximum possible power output of the entire VPP over time. The goal of the VPP is to try to keep the maximum power output above the target in the case of possible failures. As observed in the figure, both baseline and CDT fail to keep the maximum power output above the target through the time period, however, compared to the baseline, the worst case of power output capability of CDT is significantly higher, and the total time of power insufficiency is shorter than baseline. This is achieved due to similar reasons in the ramp-up scenario, since the DTs in the systems could be aware of the upcoming failure in the wind DERs, and try to use the energy storage unit to balance the power insufficiency.

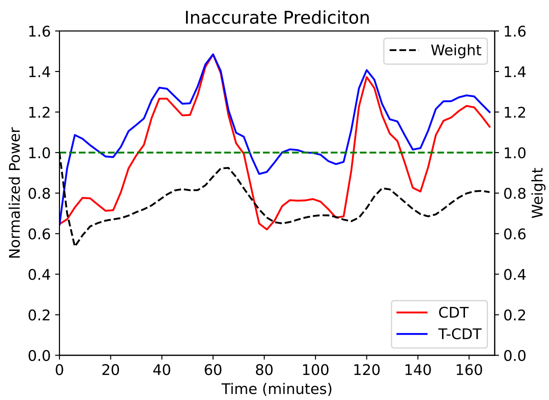

Figure 5: Inaccurate prediction.

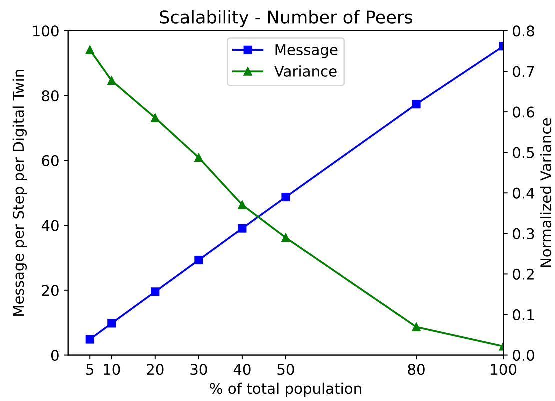

Figure 6: Changing the number of peers.

Figure 5 shows the evaluation of the effectiveness of the trust mechanism. In this figure, the collaborative DT (CDT) used in the previous experiment acts as the baseline, and is compared with trust-enabled collaborative DT (T-CDT). To gain deeper insights into the experiment's results, we additionally plot the weight assigned to the faulty DT. This weight reflects the level of trust that other DTs placed in the faulty one over time. In the scenario setup, three periods of power insufficiency were planned. Firstly, it could be observed that due to the initial power insufficiency, the faulty DT could not provide the power output that it had predicted. Thus, the weight is rapidly reduced in all other instances, causing the decision makers in each DT to leave out more safety margins. When overall power output exceeds demand, the weight assigned to the faulty DT increases. This occurs because the faulty behavior is no longer readily apparent under these conditions. By incorporating a trust mechanism that adjusts the weight of contributions from individual DTs, we effectively mitigated the second period of insufficiency and prevented a third one, as observed in the figure.

To evaluate the scalability of the framework, another set of experiments is conducted. These experiments focus on two key metrics: the communication overhead and the accuracy of the predictions. The communication overhead is measured by the number of messages per simulation timestep per DT, and the performance of the system is measured by a normalized variance. A lower variance indicates the system has fewer fluctuations thus better performance. Also, to investigate the relationship between overhead or performance to the frequency of message exchange, we introduce a parameter called *event interval*, which indicates the interval that DTs share their predictions with other DTs. Figure 6 shows the change of communication overhead and performance on different ranges of communication. In this experiment, the number of DTs is set to 200 and the event interval to 3, and the number of peers that DTs communicate to is changing. The X axis shows the number of peers in the form of a percentage of the total population, varying from 5% (10 peers) to 100% (200 peers). The result shows that as the number of peers increases,

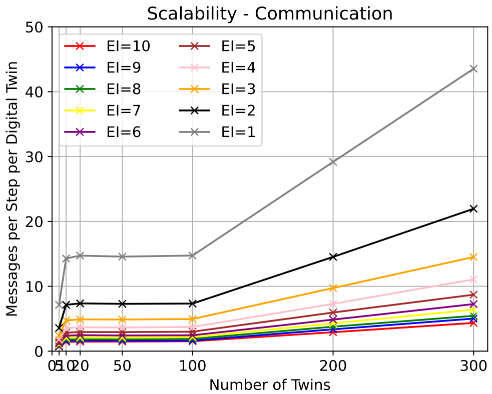

Figure 7: Overhead v.s. Number of DTs.

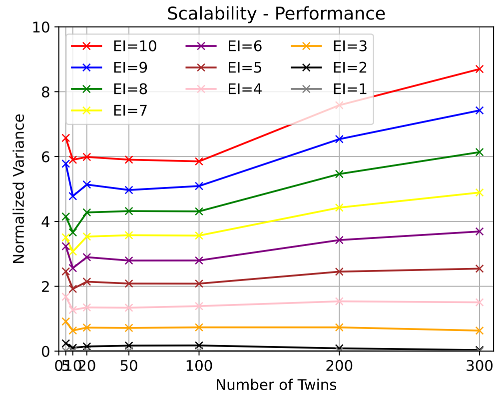

Figure 8: Performance v.s. Number of DTs.

the communication overhead increases linearly. Also, it could be observed from the figure that with the increase in the number of peers, the performance of the collaboration increases.

Figures 7 and 8 show the communication overhead and performance over different event intervals and number of DTs. In this experiment setup, the number of peers each DT communicates is the maximum value between 10% of the total population and 10. Therefore, the number of peers of each DT is at most 10. We could observe from the figures that as the number of DTs changes from 5 to 10, the communication overhead improves rapidly, because both the number of peers and the total number of DTs are increasing. From 10 to 100, since the number of peers remains at 10, the number of messages per step per twin remains constant. Starting from 100, the number of messages begins to increase again as the number of peers is growing, but still keeps a linear growth. We could also see that the event interval affects the number of messages in a predictive way.

Looking at the performance of the system over different event intervals and the number of twins, we can conclude that the event interval is an important factor in the performance. However, since the event interval also heavily affects the communication overhead, a balance between the communication overhead and the performance needs to be found. Looking at the performance as the number of twins grows, we see that the performance gets better when the number of twins changes from 5 to 10, and gradually gets worse as the number of twins increases. The reason behind the trend between 5 and 10 instances is that the capability of adjustment of 5 instances is not enough to cope with the experiment test case, although they have all-to-all communication. As the number of twins increases to 20, 50, and 100, since the number of peers does not change, the performance stays unchanged. Combined with the observation in Figure 6, we can conclude that the number of peers has the greatest impact on performance. In the interval between 100 and 300, as the number of total twins increases, the percentage of peers stays constant, thus the performance slightly decreases with the growth of the population.

The evaluation results show that the proposed framework successfully boosted the system performance over the baseline case on two common scenarios, and the trust mechanism improved the system's robustness over bad predictions. The evaluation of the scalability shows the framework's potential to scale up while maintaining the performance and keeping the communication overhead in a reasonable range.

5 CONCLUSION AND FUTURE WORK

In this paper, a new approach to enable collaboration in a multi-DT system is discussed. The proposed framework is designed to work on a high level of information exchange, which does not require complicated synchronization across different instances or sharing private data. A set of experiments is conducted with an example of the virtual power plant, and the results show that the proposed framework improves the overall performance of the system, and with the trust mechanism, system robustness is also improved. The experiment on scalability shows that the framework works well on systems of different scales, and could scale to a larger system with acceptable overheads.

Future research will focus on improving the local optimum of the participating DTs. One of the future directions of this work is the refinement of the communication protocol. Currently, the communication procedure is one-way, meaning that the recipient is only made aware of the decisions and predictions, but cannot negotiate the decision. Research is needed to develop a negation procedure in the decision exchange, which will improve the local optimum of DTs, and therefore practically encourage stakeholders to participate in the collaboration network.

ACKNOWLEDGMENTS

This research is supported by the SUSTech Research Institute for Trustworthy Autonomous Systems (RITAS) and the SUSTech-University of Leeds Joint PhD Programme. Georgios Theodoropoulos is the corresponding author of this article.

REFERENCES

Diamantopoulos, G., R. Bahsoon, N. Tziritas, and G. Theodoropoulos. 2023. “SymBChainSim: A Novel Simulation Tool for Dynamic and Adaptive Blockchain Management and its Trilemma Tradeoff”. In *Proceedings of the 2023 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*, SIGSIM-PADS ’23, 118–127. New York, NY, USA: Association for Computing Machinery https://doi.org/10.1145/3573900.3591121.

Diamantopoulos, G., N. Tziritas, R. Bahsoon, and G. Theodoropoulos. 2022. “Digital Twins for Dynamic Management of Blockchain Systems”. In *2022 Winter Simulation Conference (WSC)*, 2876–2887 https://doi.org/10.1109/WSC57314.2022.10015447.

Diamantopoulos, G., N. Tziritas, R. Bahsoon, and G. Theodoropoulos. 2024. “Dynamic Data-Driven Digital Twins for Blockchain Systems”. In *Dynamic Data Driven Applications Systems*, edited by E. Blasch, F. Darema, and A. Aved, 283–292. Cham: Springer Nature Switzerland.

Fang, X., S. Misra, G. Xue, and D. Yang. 2011. “Smart grid—The new and improved power grid: A survey”. *IEEE communications surveys & tutorials* 14(4):944–980.

García-Valls, M. and A. M. Chirivella-Ciruelos. 2024. “CoTwin: Collaborative improvement of digital twins enabled by blockchain”. *Future Generation Computer Systems*.

Ghandar, A., A. Ahmed, S. Zulfiqar, Z. Hua, M. Hanai and G. Theodoropoulos. 2021. “A Decision Support System for Urban Agriculture Using Digital Twin: A Case Study With Aquaponics”. *IEEE Access* 9:35691–35708 https://doi.org/10.1109/ACCESS.2021.3061722.

Grieves, M. and J. Vickers. 2017. “Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems”. *Transdisciplinary perspectives on complex systems: New findings and approaches*:85–113.

Hui, Y., X. Ma, Z. Su, N. Cheng, Z. Yin, T. H. Luan *et al*. 2022. “Collaboration as a Service: Digital-Twin-Enabled Collaborative and Distributed Autonomous Driving”. *IEEE Internet of Things Journal* 9(19):18607–18619 https://doi.org/10.1109/JIOT.2022.3161677.

Ismagilova, E., L. Hughes, N. P. Rana, and Y. K. Dwivedi. 2022. “Security, privacy and risks within smart cities: Literature review and development of a smart city interaction framework”. *Information Systems Frontiers*:1–22.

Lees, M., B. Logan, R. Minson, T. Oguara and G. Theodoropoulos. 2005. “Modelling Environments for Distributed Simulation”. In *Environments for Multi-Agent Systems*, edited by D. Weyns, H. Van Dyke Parunak, and F. Michel, 150–167. Berlin, Heidelberg: Springer Berlin Heidelberg.

Logan, B. and G. Theodoropoulos. 2000. “Dynamic Interest Management in the Distributed Simulation of Agent-Based Systems”. In *Proceedings of 10th AI, Simulation and Planning In High Autonomy Systems Conference*, 45–50.

Lombardo, A. and C. Ricci. 2022. “Digital twins federation for remote medical care of de-hospitalized patients”. In *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, 718–723. IEEE.

McMahan, B., E. Moore, D. Ramage, S. Hampson, Arcas and Aguera. 2017. “Communication-efficient learning of deep networks from decentralized data”. In *Artificial intelligence and statistics*, 1273–1282. PMLR.

Moshrefzadeh, M., T. Machl, D. Gackstetter, A. Donaubauer and T. H. Kolbe. 2020. “Towards a distributed digital twin of the agricultural landscape”. *Journal of Digital Landscape Architecture* 5:173–118.

Pang, J., Y. Huang, Z. Xie, J. Li and Z. Cai. 2021. “Collaborative city digital twin for the COVID-19 pandemic: A federated learning solution”. *Tsinghua Science and Technology* 26(5):759–771 https://doi.org/10.26599/TST.2021.9010026.

Pudjianto, D., C. Ramsay, and G. Strbac. 2007. "Virtual power plant and system integration of distributed energy resources". *IET Renewable power generation* 1(1):10–16.

Sahal, R., S. H. Alsamhi, K. N. Brown, D. O'Shea and B. Alouffi. 2022. "Blockchain-based digital twins collaboration for smart pandemic alerting: decentralized COVID-19 pandemic alerting use case". *Computational Intelligence and Neuroscience* 2022.

San, O., S. Pawar, and A. Rasheed. 2023. "Decentralized digital twins of complex dynamical systems". *Scientific Reports* 13(1):20087.

Suryanarayanan, V. and G. Theodoropoulos. 2013. "Synchronised range queries in distributed simulations of multiagent systems". *ACM Trans. Model. Comput. Simul.* 23(4) https://doi.org/10.1145/2517449.

Ventresque, A., Q. Bragard, E. S. Liu, D. Nowak, L. Murphy, G. Theodoropoulos *et al*. 2012. "SParTSim: A Space Partitioning Guided by Road Network for Distributed Traffic Simulations". In *2012 IEEE/ACM 16th International Symposium on Distributed Simulation and Real Time Applications*, 202–209 https://doi.org/10.1109/DS-RT.2012.37.

Vergara, C., R. Bahsoon, G. Theodoropoulos, W. Yanez and N. Tziritas. 2023. "Federated Digital Twin". In *2023 IEEE/ACM 27th International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*, 115–116 https://doi.org/10.1109/DS-RT58998.2023.00027.

Villalonga, A., E. Negri, L. Fumagalli, M. Macchi, F. Castaño and R. Haber. 2020. "Local decision making based on distributed digital twin framework". *IFAC-PapersOnLine* 53(2):10568–10573.

Waldrop, M. M. 1993. *Complexity: The emerging science at the edge of order and chaos*. Simon and Schuster.

Zhang, N., R. Bahsoon, and G. Theodoropoulos. 2020. "Towards Engineering Cognitive Digital Twins with Self-Awareness". In *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 3891–3891 https://doi.org/10.1109/SMC42975.2020.9283357.

Zhang, N., R. Bahsoon, N. Tziritas, and G. Theodoropoulos. 2024a. "Explainable Human-in-the-Loop Dynamic Data-Driven Digital Twins". In *Dynamic Data Driven Applications Systems*, edited by E. Blasch, F. Darema, and A. Aved, 233–243. Cham: Springer Nature Switzerland.

Zhang, N., R. Bahsoon, N. Tziritas, and G. Theodoropoulos. 2024b. "Knowledge Equivalence in Digital Twins of Intelligent Systems". *ACM Trans. Model. Comput. Simul.* 34(1) https://doi.org/10.1145/3635306.

AUTHOR BIOGRAPHIES

ZHENGCHANG HUA is a PhD candidate in the SUSTech-University of Leeds Joint PhD programme. His research interests include digital twin networks, agent-based systems, and discrete event simulation. He holds a BSc in Computer Science from the Southern University of Science and Technology, China. His email address is huazc@mail.sustech.edu.cn.

KARIM DJEMAME is a Professor of distributed systems in the School of Computing at the University of Leeds, UK. His research interests are in the areas of distributed systems; cloud computing; energy efficiency; heterogeneous parallel architectures; performance evaluation. He holds a PhD from the University of Glasgow, UK. His email address is K.Djemame@leeds.ac.uk.

NIKOS TZIRITAS is an Associate Professor with the Department of Informatics and Telecommunications at the University of Thessaly, Greece, where he also obtained his PhD. Prior to this he was an Associate Professor at the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China. He is the recipient of the 2016 Award for Excellence for Early Career Researchers in Scalable Computing from IEEE Technical Committee in Scalable Computing. His email address is nitzirit@uth.gr.

GEORGIOS THEODOROPOULOS is currently a Chair Professor at the Department of Computer Science and Engineering at SUSTech in Shenzhen. In the past he has held senior appointments at Universities of Durham and Birmingham in the UK, IBM Research, Ireland, and Nanyang Technological University, Singapore. He holds a Ph.D. from the University of Manchester, UK. He is a Member of the European Academy of Sciences and Arts and a Fellow of the World Academy of Art and Science. His email address is theogeorgios@gmail.com.