Journal of Computer Science

Original Research Paper

Radial Basis Function Neural Networks Towards Electronic Trust Quantification

[1,2]Smitha Zacaria and [2]Tiny Du Toit

[1]Department of Timetables and Assessments, North-West University, Potchefstroom, South Africa
[2]Unit for Data Science and Computing, North-West University, Potchefstroom, South Africa

Article history
Received: 29-04-2024
Revised: 12-08-2024
Accepted: 24-08-2024

Corresponding Author:
Smitha Zacaria
Department of Timetables and Assessments, North-West University, Potchefstroom, South Africa
Email: smitha.zacaria@nwu.ac.za

Abstract: Trust is a broad term applicable in various contexts. Trust between electronic entities is complex to quantify, particularly in intricate networks. Traditional trust algorithms rely on historical trust values, requiring storage and access to past transaction details. Contextual variations further complicate trust calculation. Challenges in calculating trust include heightened computational complexities, managing information storage, and securing access to extensive datasets crucial for evaluation. This study will explore the implementation of a Radial Basis Function Neural Network (RBFNNet) to evaluate trust. This neural network effectively approximates functions, addressing complex computational challenges. Its efficacy depends on ample training data for modeling trust values. Synthetic data generation becomes crucial to surmount the scarcity of trust datasets. This study introduces a new definition of trust between electronic entities, a seven-step Pure Synthetic Trust Data Set Generation (PSTDG) framework for generating artificial trust datasets, a new definition for validating generated trust data, and a method with three steps to authenticate the model for generating data. A process consisting of four steps was developed to design an RBFNNet model for determining trust values between electronic entities. Two experiments were conducted to determine trust values using the RBFNNet. A trust dataset was generated using the PSTDG-PeerTrust model, which incorporates the fundamental principles of trust assessment outlined in the PeerTrust model. The validation process was completed, and a new RBFNNet model PeerTrustRBFNNet was developed, utilizing optimal hyperparameters. During the second phase of experimentation, the Amazon Relational Database Service was used to exhibit the efficiency of the proposed approach in tackling real-world problems. The study determined that the PSTDG framework-generated models create valid synthetic trust datasets, and an RBFNNet effectively computes trust in digital environments. Moreover, novel definitions of trust and synthetic trust dataset validation were developed, contributing to the advancement of trust assessment methodologies in various contexts.

Keywords: Data Generation, Electronic Trust, Synthetic Trust Data, Synthetic Trust Data Generation, Synthetic Trust Data Validation

Introduction

In the rapidly evolving digital landscape, the concept of trust is paramount, especially when it comes to interactions between electronic entities such as software systems, applications, and online services. Trust is a multifaceted concept that is fundamental to various domains, including service delivery, credibility, information security, and reliability (Alawneh and Abbadi, 2022; Singal and Kohli, 2016; Olmedilla *et al.*, 2006; Grandison and Sloman, 2003; Gambetta, 2000). Many researchers provide various definitions for the concept of trust (Zacaria, 2023). However, each definition is relevant only within its specific context and assumes the measurability of trust. For instance, online banking services are trusted to secure financial information and downloaded apps are trusted to perform as advertised without compromising data.

Interactions become more dependable when grounded in trust (Yong-Sheng and Ying, 2010). Quantifying trust between electronic entities presents significant challenges, particularly within complex networks. Traditional trust algorithms rely heavily on historical data, requiring extensive data storage and access (Tahta *et al.*, 2015; Li and Ling, 2002; Aberer and Despotovic, 2001). This complexity is further compounded by contextual variations, highlighting the need for innovative methods to quantify trust. Trust is not symmetric, as it depends on the ability to act within a specific context (Sagar *et al.*, 2024; Alawneh and Abbadi, 2022; Grandison and Sloman, 2003). This implies that the level of trust Entity V places in Entity W is different from the trust Entity W has in Entity V. Trust in an entity varies by context. For example, trust in an entity in the context of service delivery may not match trust in the context of credibility. Most existing trust assessments focus on information security. No standard list of contexts under the field of information technology and computer science could be found.

Trust is essential in numerous real-world scenarios. Consider using smart contracts in block chain technology, where trust between decentralized applications ensures that transactions are executed as programmed without third-party interference. In online marketplaces like Amazon or eBay, trust determines whether buyers feel confident purchasing from sellers, relying on reviews and ratings as indicators. Another example is in autonomous vehicles, where trust between software components ensures that data from sensors is accurately processed to make safe driving decisions. Trust in medical software systems is crucial for accurate diagnosis and treatment in healthcare, ensuring patient data is handled securely and accurately. Within corporate environments, trust between different software applications allows seamless data integration and workflow automation, enabling efficient business processes. Across all these scenarios, having a reliable method to calculate and maintain trust can significantly impact the trustworthiness of the systems.

As the digital world continues to grow and evolve, the need for precise and adaptable trust evaluation mechanisms becomes more critical, underscoring the importance of advanced computational models. Developing innovative methods to quantify and manage trust in these diverse contexts is essential for a more interconnected and trustworthy digital future. Despite trust's criticality in technology-driven systems, achieving a universally accepted definition applicable across all computational contexts remains elusive, reflecting a significant research gap. This study aims to address this gap by proposing a refined definition of trust tailored specifically to electronic environments. Beyond definitional challenges, quantifying trust presents practical impediments. Existing assessments primarily focus on information security contexts, necessitating diverse trust evaluation approaches adopting varied architectures for computing and managing trust values. However, increasing contextual complexity and network intricacies amplify the challenges of trust value calculations.

Certain entities lack direct knowledge or experience from past transactions with other entities in a decentralized system. Algorithms designed to compute trust rely on these entities' historical trust values to calculate new ones. To accurately assess the trustworthiness of each entity relative to others, it is crucial to store and access detailed historical interaction data and trust scores within the network. Additionally, there is a significant research gap related to the increasing complexity of trust calculations, which is compounded by the intricate nature of electronic entity networks. Addressing this challenge, this study leverages the unique capabilities of the RBFNNet approach to quantify trust. The study outlines a four-step method for developing RBFNNet-based trust calculation models, highlighting their proficiency in approximating complex functions to mitigate inherent computational challenges. However, their efficacy depends on sufficient training data, posing a challenge due to the scarcity of trust datasets. To address this challenge, the study introduces the PSTDG framework, a comprehensive seven-step process designed to generate validated trust datasets across diverse contexts. This framework not only establishes a foundation for training robust RBFNNet models but also addresses the gap between data scarcity and model efficacy.

A pivotal aspect of the PSTDG framework is the rigorous validation of synthetic trust datasets. This validation process introduces a novel definition for assessing synthetic trust datasets, employing a three-step validation approach. By ensuring the validity of synthetic trust data, this framework enhances the development of precise trust calculation models critical for advancing trust evaluation in complex decentralized networks. The practical application of the PSTDG framework is demonstrated through the development and validation of two PSTDG models, specifically PSTDG-PeerTrust and PSTDG-ARDS. In this study, the models named PeerTrustRBFNNet and ARDSTrustRBFNNet are trust calculation models developed using an RBFNNet. The study successfully implements an alternative approach to trust computation utilizing the RBFNNet.

Trust between Electronic Entities

In today's digital age, trust gains added relevance,

particularly with the internet's pervasive influence on various aspects of human existence. As millions engage in daily online transactions, data mismanagement poses significant risks, prompting the crucial question of how trust is established in the virtual space, especially between electronic entities. In this section, trust between electronic entities will be defined. The approach employed involves two steps. Firstly, a comprehensive review of the literature explores various definitions of trust. Secondly, commonalities and differences are identified, leading to the recognition of three dimensions of trust and a novel, comprehensive definition (Zacaria, 2023).

Definition of Trust

Mui *et al.* (2002) proposed a computational model of trust and reputation based on their study. They defined reciprocity as a mutual interchange of actions and described reputation as an agent's belief shaped by previous behaviors. The research proposes that trust is subjective and based on previous interactions with an entity. Gambetta (2000) defined trust as a subjective evaluation of the anticipated performance of an agent.

Hang *et al.* (2012) studied service selection based on trust in composite services, offering a method to assess trust in individual components. In addressing the challenge of similar services, they emphasized that customers evaluate required functionalities and Quality of Service (QoS). QoS, subjective and dependent on preferences, requires exchangeable information and dynamic monitoring. In service-oriented systems, trust is described as a probability-driven assessment of service quality.

In 2000, an attempt was made by Witkowski and Pitt established a trust representation for software agents, characterizing it as the extent of reliance V places on W within the framework of selecting agents in a context involving multiple autonomous agents. This involves the selection of agents within a trading community comprising multiple agents (Witkowski and Pitt, 2000).

Au *et al.* (2001) introduced a trust paradigm for cross-organizational interactions on extranets, which are extensions of intranets that allow external user access. The authors identify extranet types, trust management requirements, and infrastructure approaches. Their proposed framework distributes trust on extranets across organizations, detailing protocols for establishing trust within the network of trust and includes an automated system for trust derivation and composition.

Wang and Vassileva (2003) introduced a trust framework utilizing Bayesian networks, describing an approach to construct reputation within Peer-to-Peer (P2P) networks through endorsements. They outlined definitions and attributes of trust and reputation, highlighting that trust is formed through firsthand interactions between two entities.

After surveying various definitions, Grandison and Sloman (2003) established a definition for trust, specifically in the context of internet applications. They discussed trust relationship properties, different types of trust, and trust management. Emphasizing the context-specific nature of trust, they noted that a trustor-trustee relationship is not absolute, relying on the ability to perform specific tasks within certain circumstances. Trust exhibits non-symmetry and non-transitivity, involving various classes, such as infrastructure trust, trustee certification, entrustment, trust in resource access, and trust in service provision. Trust is earned through secure and reliable dealings. For internet applications, they created SULTAN, a Simple Universal Logic-oriented Trust Analysis Notation. SULTAN offers capabilities for specifying, analyzing, and managing trust connections. The parameters of trust connections are established by a sequence of steps, each assigned a trust level, along with conditions that need assessment for the trust relationship to be applicable.

The interpretation of trust varies, depending on the context, with a focus on policy-based and reputation-based trust as the primary methods for managing trust, especially in the lifecycle of virtual organizations and the control of resource access in grid computing (Olmedilla *et al.*, 2006).

Mohsenzadeh and Motameni (2015) introduced TMFM, a fuzzy mathematics-based trust model for the cloud computing environment. They emphasized the complexity of trust relationships, noting their subjectivity, non-symmetry, partial transitivity, dynamic nature, context-dependency, uncertainty, and evaluation challenges. The model calculates fuzzy direct trust relations among cloud entities based on direct experiences, defining the subjective probability of trust as an entity's ability to accomplish a task within a designated time frame and context, guided by recommendations from trustworthy entities.

Das and Debnath (2020) elucidate that trust within computer networks is the expectation of one node towards another, establishing a relational foundation based on predefined criteria and contextual factors. This mutual trust mechanism bolsters the confidence of nodes prior to engaging in communication exchanges.

A survey focusing on trust evaluation based on machine learning was conducted by Wang *et al.* (2021). This survey shed light on trust evaluation as a process

aimed at quantifying trust using various influential attributes. However, this evaluation process encountered significant challenges, including the scarcity of essential evaluation data, the need for extensive big data processing, the demand for simplified expressions of trust relationships, and the expectation of automation. These obstacles underscore the complexity and pressing need for advancement in trust evaluation methodologies.

Lu *et al.* (2023) defined trust in the context of open microservices systems as the belief held by a truster in a trustee, wherein the trustee is expected to provide or accomplish the services it claims it will provide and satisfy the truster's expectations within a specific context for a designated period of time.

Trust Dimensions

The definitions of trust discussed above will be analyzed in this section to identify the dimensions of trust, considering both similarities and differences. Researchers often tie trust to specific contexts, such as reputation, service selection, or internet applications. Context, a key factor in trust assessment, varies across different trust definitions. Notably, trust is considered context-dependent by various authors Lu *et al.* (2023); Zacaria (2023), and its versatility allows effective application in diverse settings, such as security, service delivery, reliability, and credibility.

Trust can be quantified as a probability, as highlighted in definitions provided by Das and Debnath (2020); Mohsenzadeh and Motameni (2015); Hang *et al.* (2012); Gambetta (2000). Various trust models employ different algorithms for quantification, especially in the security context (Mohsenzadeh and Motameni, 2015; Wang *et al.*, 2008; Wang and Chi, 2006; Au *et al.*, 2001). The quantification process considers parameters, such as historical encounters, set standards, and direct experiences, reflecting the multifaceted nature of trust assessment.

Trust establishment involves retrieving information from various origins (Wang *et al.*, 2021; Hang *et al.*, 2012; Wang and Vassileva, 2003; Mui *et al.*, 2002; Au *et al.*, 2001; Gambetta, 2000; Witkowski and Pitt, 2000; Denning, 1993). The subjective nature of trust emphasizes the need for information that can be obtained from customers, network peers, given standards, and other sources. In a P2P network, transaction feedback is a key information source, utilizing parameters, such as peer trust, satisfaction, transaction frequency, and size. Websites assess user behavior, using parameters, such as time spent and page visits to enhance information confidence. Entities with given standards combine information from those standards with customer satisfaction feedback for trust calculation. Increasing parameters in trust calculation further fortify confidence by capturing a broader spectrum of relevant factors, ensuring a robust and adaptable assessment.

As discussed above, trust encompasses three dimensions: The context, calculation or quantification, and information sources for the calculation. The required information depends on the algorithm and context.

New Definition of Trust

Building on trust definitions and subsequent insights, it is evident that trust is often contextualized in its application. However, its generic nature allows broad applicability. The probabilistic calculation of trust is underscored, emphasizing the need for diverse information sources. Recognizing the need for a versatile definition, three vital dimensions are identified. Informed by these dimensions, trust is defined as follows, acknowledging its multi-dimensional essence shaping trust dynamics: "Trust is a quantified belief or a probability of belief of an entity, which can be calculated by accessing or using information from different sources which are based on some set of standards or guidelines, to have some desired property within a specified context" (Zacaria, 2023).

Trust Calculation Using Neural Networks

Determining trust values is a complex and time-consuming task due to various factors, encompassing network complexity, information access, data management, data storage, and context complexity. Feed-forward neural networks include multi-layer perceptrons and RBFNNets. An RBFNNet offers a simpler and faster alternative to multi-layer perceptron networks (Shakya *et al.*, 2011). With a straightforward training process and improved resilience to input disturbances, an RBFNNet effectively handles patterns not included in the training dataset, making them widely applied across various domains. The RBFNNet is highly regarded for its ability to learn more quickly compared to other feed-forward networks. This feature positions it as a well-suited model to tackle approximation challenges (Ruslan *et al.*, 2013; Chien-Cheng *et al.*, 1999). RBFNNets are known for their simplicity in topology structure and explicit learning process. They are effective function approximators, especially when it comes to solving non-linear problems (Lin and Wu, 2011). Broomhead and Lowe (1988) showed how an RBFNNet can transform non-separable linear problems into separable ones. RBFNNets find extensive application in various domains, encompassing tasks such as the identification of systems with non-linear characteristics and forecasting within the context of temporal analysis (Moody and Darken, 1989; Broomhead and Lowe, 1988). Consequently, it is selected as the trust calculation technique proposed in this study.

The primary objective of this research is to create another approach for trust computation, employing an RBFNNet that can help resolve some of the issues mentioned earlier. To achieve this goal, the study will focus on two primary objectives:

1. Generating training data
2. Building an RBFNNet trust model

After the completion of the initial objective, the issue of not having an extensive dataset to use as training data will be resolved. Therefore, the second objective is building an RBFNNet trust model. In the next section, the challenges in collecting trust data will be discussed.

Challenges in Collecting Trust Data

In dynamically shifting scenarios, building trust values for electronic entities through training an RBFNNet demands sizable datasets. To leverage diverse data sources, conventional algorithms encompass prior experiences, human behavior, recommendations, feedback, and adherence to W3C methodologies for trust calculation (Al-Shargabi, 2016; Singal and Kohli, 2016; Tahta *et al.*, 2015; Xiong and Liu, 2004; W3C, 2002). An architectural framework is imperative for submitting and retrieving feedback, trust value computation, and trust value management. However, various systems may adopt distinct data storage and retrieval methods from past transactions, which may introduce complexities in this process (Xiong and Liu, 2003).

The utilization of trust-related training data poses several challenges. The confidentiality of the data makes it difficult to acquire from customers and suppliers. Acquiring data for conventional algorithms and suggestions can be arduous, and the absence of previous interaction experience hinders trust assessment when interacting with a software component or program for the initial interaction. Inaccurate feedback is a potential concern, influencing trust values. Accessing stored data may impact the speed and calculation time of the trusted network. Consequently, collecting data for trust calculation can be costly and time-consuming, with privacy concerns limiting trust data disclosure.

To overcome data scarcity, traditional algorithms simulate real-world data (Xiong and Liu, 2003-2004; Thacker *et al.*, 2004; Wang and Vassileva, 2003; Aberer and Despotovic, 2001). However, this study aims to generate synthetic data to supplement the real-world dataset and address data scarcity.

Synthetic Data Generation and Validation

One way to address the limited availability of real-world training data is by programmatically creating artificial data (Anderson *et al.*, 2014). It has been extensively studied and effectively utilized in various fields, including forecasting the cash availability at ATMs (Ranja *et al.*, 2023), visually identifying leaks in industrial systems (Gitzel *et al.*, 2021), traffic generation (Bhaumik *et al.*, 2015), renewable energy technology performance modeling (Pillai *et al.*, 2014), building deterioration modeling (Scheidegger and Maurer, 2012), fraud detection (Barse *et al.*, 2003), and more. Realism is a significant aspect to consider when it comes to the caliber of a synthetic dataset (Tsvetovat and Carley, 2016). Datasets of this kind should accurately represent real-world data and meet certain standards to ensure their accuracy.

Synthetic Trust Data Validation and the Definition of Validation

Ensuring the precision of the synthetic trust data generation model is crucial, as inaccuracies can compromise the reliability of the RBFNNet model for predicting trust. Validation entails examining the model's results, particularly the trust dataset it produces.

Rykiel (1996) defined validation as determining whether an ecological model meets performance requirements for its intended use. According to Carley (1996), validation in computational modeling pertains to the procedures and methods used to ensure that the generated or simulated data aligns with genuine data. The author characterizes authentic data as information obtained via experimentation, fieldwork, archival research, or survey analyses. Sargent (2020) outlines validation as the procedural aspect of ascertaining the accuracy of a model's outcomes within its intended scope. Thacker *et al.* (2004) delineate a procedure for amassing supporting proof to showcase the accuracy of a model within its designated application. Kleijnen (1998) characterizes validation as a systematic procedure to verify if a simulation model faithfully mirrors the actual system. Describing validation as a method to evaluate the accuracy of a calculation approach within its designated real-world application emphasizes the importance of ensuring reliability in practical scenarios (Jones *et al.*, 2004). Basile and Ferrara (2023) underscore that validating systems involves ensuring compliance with requirements through rigorous testing, particularly for critical applications, which can be resource-intensive. According to Law (2019), validation is elucidated as a procedure for establishing whether a model accurately portrays the real world in alignment with the study's planned aim or goals.

New Definition of Validation

Validation, as a methodical process, exhibits varying interpretations shaped by the distinct needs of users or researchers. This crucial step involves verifying the model's precision in faithfully mirroring real-world conditions, as highlighted by researchers

such as Basile and Ferrara (2023); Thorve *et al*. (2022); Law (2019); Jones *et al.* (2004); Kleijnen (1998); Carley (1996). The overarching goal of validation is to furnish compelling evidence attesting to the model's accuracy, ensuring it aligns sufficiently with its intended use, as emphasized by Thacker *et al.* (2004).

Within the domain of generated trust data, the definition of validation is formulated as follows, drawn from the previous discussion: "Validation is a process or method to provide evidence that can be used to determine whether the synthetic trust data generation model is generating a valid trust dataset under its specific scenarios for its intended use" (Zacaria, 2023).

Sensitivity and What-if Analyses

Validating synthetic data against real-world data is crucial for accuracy. Depending upon the availability of real data, various validation methods are used, such as statistical validations, graphical plots, Schruben-Turing tests, and Sensitivity and What-if Analyses (SWA) (Law, 2019; Kleijnen, 1998). Effective validation of most of these techniques necessitates the availability of real-world data. This research verifies a PSTDG model, elaborated in the upcoming section, by applying the novel validation definition presented in the previous section. Owing to the limited availability of trust data from real-world sources, conventional validation methods become impractical. In place of these, qualitative expert knowledge is used to determine how input variables affect the synthetic trust data generation model's output behavior. The validation process involves checking whether the model aligns with this expert knowledge (Kleijnen, 1998). When real-world data is unavailable, resorting to an SWA proves valuable for the validation process.

SWA investigates input-output relationships in a model (Nguyen *et al.*, 2018; Chan *et al.*, 2010; Kleijnen, 1997). The sensitivity analysis examines the impact of extreme values, while the what-if analysis explores the effects of alterations to parameters or variables. Multiple runs are conducted to account for changing factors across different scenarios by varying specific input factors while keeping others constant. This study uses SWA to validate the trust dataset generated by the simulation model. There are various ways to validate data, such as mathematical, statistical, and graphical methods (Christopher Frey and Patil, 2002). Graphical methods visually represent trust data changes with different inputs. The ultimate aim is to create a reliable trust calculation method using an RBFNNet without comparing it to other methods. Graphical methods provide sufficient validation for the dataset.

Scatter plots are widely recognized for their simplicity and efficacy in aiding comprehension of the association between factors (Chan *et al.*, 2010; Kleijnen, 1997). This study will use a scatter plot to display model output on the z-axis and one or two different input variables on the x- and y-axes, respectively.

SWA of the trust dataset will be done through the following steps:

1. Compile what-if scenarios based on expert knowledge of trust calculations
2. Generate the trust dataset and create scatter plots for each scenario
3. Analyze the scatter plots to evaluate the consistency between the visual representation of trust data changes and expert knowledge

If the generated graphs align with the expert's expectations, the data generation model is considered valid or suitable for its designated purpose. Hence, this procedure functions as a means of validation according to the new definition provided in the section above.

PSTDG Framework

The PSTDG framework given in Fig. (1) will be employed to produce a Pure Synthetic Trust Dataset (PSTDS) for assessing trust in this research. It is a structured approach that involves three stages: PSTDG model development, validation, and PSTDS generation. The first step of the framework is to determine why it is better to use a PSTDS instead of a real-life trust dataset. Thereafter, the second step involves collecting all relevant knowledge, which includes expert insights on trust calculation. Rules or guidelines function as constraints in the PSTDG model (Zacaria, 2023).

To create the trust data generation model, it is important to compile these constraints in a step called "Compile". In this step, the emphasis is on identifying how the expert knowledge uncovered in the second step has the potential to be utilized in computing the trust value. Following the compilation of constraints, it is necessary to decide on calculating the trust values and implementing the constraints based on specific needs or situations. In the PSTDG framework, the "Develop" stage involves creating a model for generating trust data. The development of an appropriate algorithm or program for the PSTDS generation must be done. The validation step aims to determine if the PSTDG model produces a valid trust dataset. The section titled "Sensitivity and What-If Analyses" explains how to validate a PSTDG model.

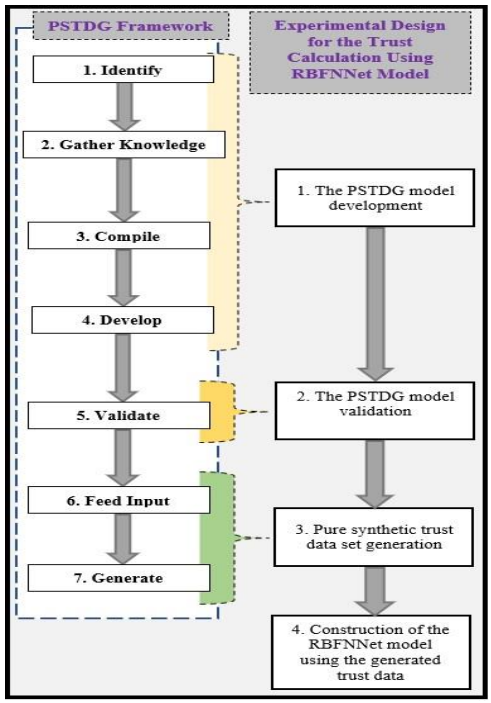

Fig. 1: PSTDG framework

The model or system for generating a PSTDS must be able to create data based on particular circumstances or defined parameter values. In the "Feed Input" step, the input parameter values required for generating a PSTDS need to be determined. The last stage in the PSTDG framework is the generation of a PSTDS.

The chosen constraints, rules, and input parameter values in Steps 3 and 6, respectively, will guide the creation of the synthetic trust dataset.

Materials and Methods

The construction of a model that utilizes an RBFNNet to discern trust levels within electronic entities, a detailed process comprising four sequential steps, is undertaken as depicted in Fig. (1). The interconnection between the developed PSTDG framework and this experimental design process is visually depicted in the same figure. The four consecutive stages within this experimental design procedure unfold in the following manner: Firstly, the development of a PSTDG model ensues, employing the initial four steps outlined in the PSTDG framework (Fig. 1). Subsequently, the PSTDG model undergoes validation, utilizing the recently introduced definition of validation explained in the preceding section. This validation process incorporates the fifth step in the PSTDG framework and will be extensively discussed in the forthcoming sections. Continuing, the approach to creating a PSTDS using the validated PSTDG model is clarified, encompassing the subsequent steps in the PSTDG framework. Finally, the development of the suggested RBFNNet model designed for trust calculation is detailed, outlining the utilization of the generated PSTDS in subsequent sections.

Two experiments were conducted in this study. In the first experiment, a trust model named PeerTrustRBFNNet is formulated, utilizing the trust model introduced by Xiong and Liu (2004) through the process structured in four steps. The objective of the second experiment was to illustrate the practicability of the proposed solution in real-world applications. To this end, the experiment utilized the Amazon Relational Database System (ARDS) (Amazon RDS Service Level Agreement, 2024; 2019).

Experiment 1- PeerTrustRBFNNet

PSTDG-PeerTrust Model Development

Firstly, the development of a PSTDG model ensues, employing the initial four steps outlined in the PSTDG framework (Fig. 1):

1) Identification of the Necessity for SDG

To effectively train an RBFNNet trust model, it is imperative to have a significant trust dataset. This requirement was explicitly recognized in the section titled "Challenges in Collecting Trust Data," underscoring the essential need for generating a comprehensive trust dataset:

2) Expert knowledge gathering

During this phase, extensive knowledge, including qualitative expert insights into trust data, is gathered. As outlined by Xiong and Liu (2004), trust among peers in P2P electronic communities relies on five crucial elements: Satisfaction derived from peer feedback, the extent of feedback in transaction quantity, reliability of the source providing feedback, transaction context, and community context. These factors play a pivotal role in understanding and assessing the dynamics of trust within P2P electronic communities.

The following were the identified expert knowledge pertaining to the calculation of a peer's trust value using Xiong and Liu PeerTrust theoretical trust calculation model:

1. The trust value of a peer should increase, based on the satisfaction gained from interactions with other peers. This new trust value should vary directly with the level of satisfaction received
2. The trustworthiness of the interacting peers ought to scale directly with the magnitude of the transaction
3. The feedback provided by a trusted peer should carry more weight
4. The trustworthiness value needs to be computed,

considering all the feedback received previously

5. The trust value might vary, depending on the community context factor, where the community context factor includes rewarding peers for giving feedback and having digital certificates:

3) Compile constraints

As per the findings of Xiong and Liu (2004), trust among peers in P2P electronic communities can be calculated in various ways based on different contexts. Users are empowered with the freedom to opt for one of the three approaches listed below to calculate trust values. In Instance 1, the basic trust matrix can be utilized, considering only the weighted average of transaction satisfaction. Instance 2 incorporates the transaction context factor, accounting for factors such as transaction size, and Instance 3 includes both the transaction and community context factors. The latter, Instance 3, is chosen in this study for trust calculation in the PSTDG-PeerTrust model development, as it comprehensively considers all five crucial aspects identified by Xiong and Liu (2004). This choice is informed by expert knowledge 1 through 5, ensuring a holistic evaluation of a peer's trust value. The trust equation is given below:

$$T(a)=\alpha\cdot\sum_{i=1}^{I(a)} S(a,i)\cdot Cr(p(a,i))\cdot TF(a,i)+\beta\cdot\frac{F(a)}{I(a)} \quad (1)$$

where, $T(a)$ represents the trustworthiness score of peer a, $I(a)$ is the overall count of transactions peer a conducted in the recent time frame, (and Sa, I) is the quantity of satisfaction where the peer gains for the i^{th} transaction. In addition, α represents the weight assigned to the transaction context factor, while β signifies the weight assigned to the community context factor, $p(a, i)$ is the other participating peer in peer a $n's$ i^{th} transaction, $Cr(p(a, i))$ represents the credibility of the interacted peer in the i^{th} transaction, determined as a function of trust value and is given below, and the count of the feedback peer a gave to others is represented by $F(a)$.$Cr(p(a, i))$ is defined as follows:

$$Cr(p(a,i))=\frac{T(p(a,i))}{\sum_{j=1}^{I(a)} T(p(a,j))} \quad (2)$$

The subsequent section will provide detailed information on the evolution of the PSTDG-PeerTrust model, representing the fourth step in the PSTDG framework:

4) Develop PSTDG-PeerTrust model

In this experiment, Instance 3 is implemented by assigning equal importance or weight to the transaction and community context factors, ensuring comprehensive consideration of all five critical factors identified in this section. The model is executed through two stored procedures using MS SQL code. The first procedure creates peers, assigns initial trust values, and determines interaction details, while the second procedure updates trust values based on feedback, credibility, transaction details, and context factors. The transaction context involves classifying transactions by size. The latter was classified into categories, such as small, medium, large, or extra-large.

Validation of the PSTDG-PeerTrust Model

The validation process for the PSTDG-PeerTrust model followed the definition outlined in the section titled "New Definition of Validation". The validation took place during the fifth step of the PSTDG framework.

The initial phase of the validation process was to compile what-if scenarios based on expert knowledge of trust calculation. A valid dataset must satisfy all identified expert knowledge, and all what-if scenarios were compiled using this knowledge. If the visual representations created from the data align with the expected outcomes based on expert knowledge, the model for generating the data can be regarded as valid for its designated application. Thus, trust data with scatter plots were created based on the what-if scenarios. The resulting graphs were analyzed to see if trends matched the expert knowledge (Zacaria, 2023). Below is an example of a what-if analysis conducted on the PSTDG-PeerTrust model.

According to expert knowledge 1, given above, the new trust value between two peers should depend on the level of satisfaction received after an interaction. In other words, the more satisfied a peer is with the interaction, the greater their updated trust value ought to be. The following steps were taken to validate the PSTDG-PeerTrust model.

Step 1: How is the level of trust attributed to Peer 1 affected as the satisfaction level from other peers gradually rises, varying from a minimum of (0) to a maximum of (0.999) throughout a sequence of transactions (Zacaria, 2023)? Peer 1 engaged in a series of transactions with other peers. Following each transaction, the trustworthiness rating of Peer 1 was computed, considering a linear increase in the satisfaction level from the other peer while keeping all other parameters constant.

Step 2: In this scenario, the following parameters were held constant:

- The total transaction count was set at 1,000
- The overall peer count included 1,001 individuals (Peer 1-1001), with Peer 1 engaging in interactions with all other peers
- The value associated with the transaction context, specifically the transaction size, was designated as 1
- All peers commenced with an initial trust value of 1
- The weight factors, denoted as α and β, were both set to 0.5

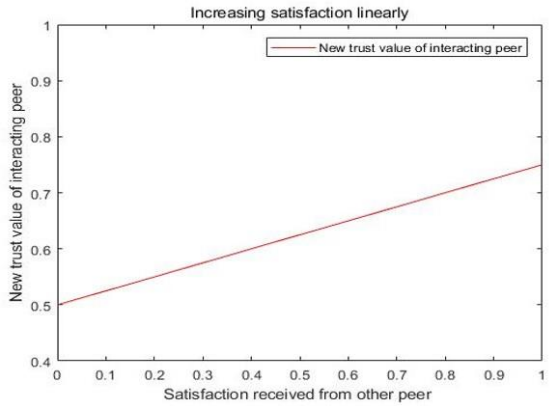

Fig. 2: Trust value when the satisfaction received is increasing linearly

As part of the transaction context factor, a random value within the specified ranges was assigned to each transaction based on size. For extra-large transactions, a randomly assigned value between 0.75 and 1.0 was used. Large transactions were assigned values from 0.5-0.75, medium transactions from 0.25-0.5, and small transactions from 0-0.25.

The plot in Fig. (2) illustrates the new trust value of Peer 1 on the y-axis plotted against the satisfaction derived from other peers on the x-axis over the course of 1000 transactions.

Step 3: The graph in Fig. (2) demonstrates the fluctuation in the trust value of the interacting peer, moving from 0.5-0.749, in response to changes in the 'satisfaction received from the other peer' factor ranging from 0-0.999 while maintaining constant values for all other parameters.

The plot displays an almost linear correlation between the "satisfaction received from the other peer" parameter and the "new trust value of the interacting peer". The trustworthiness rating of the interacting peer escalates as the "satisfaction received from the other peer" parameter goes up. Figure (2) shows that the satisfaction received directly influences the new trust value. Thus, the data satisfies the identified expert knowledge given above.

In the next section, PSTDS generation using the PSTDG-PeerTrust model will be discussed.

PSTDS Generation Using PSTDG-PeerTrust Model

Steps six and seven of the PSTDG framework involved generating a PSTDS. In total, the PSTDG-PeerTrust model generated 10000 data points with 14 input variables. These variables include the interacting peer's name, the identity of the another peer involved in the interaction, the size of the transaction, the transaction size value, an indication of whether the interacting peer received feedback from the other peer, the amount of satisfaction obtained from the other peer, an indication of whether the other peer received feedback from the interacting peer, the amount of satisfaction received from the interacting peer, the interacting peer's existing trust value, the existing trust score of the other peer, the total number of feedback given by the interacting peer, the total number of feedback given by the other peer, and the updated trust values for both interacting and other peers after the transaction.

Construction of PeerTrustRBFNNet Model

The generated PSTDS was used as a time series by implementing sliding windows. This was utilized to build the proposed PeerTrustRBFNNet model. The process involved three stages: Preprocessing the data, determining the optimal RBFNNet model, and assessing the best model. Data preprocessing can enhance the accuracy of an Artificial Neural Network (ANN) model, reduce computational costs, and expedite the learning process (Koval, 2018; Kuźniar and Zając, 2017; Nawi *et al.*, 2013). In addition, preprocessing the input variables can improve the matching of predicted output (Koval, 2018; Kuźniar and Zając, 2017). As an ANN can only process numerical data, some inputs were converted into one-hot encoded binary values. As per the requirement of the ANN, the data was normalized to the range of [0, 1] to ensure equal treatment of all values (Koval, 2018). After preprocessing, the PSTDS was randomly partitioned into training (70%), validation (20%), and test sets (10%).

To construct an accurate RBFNNet model for determining trust scores among digital entities, appropriate model hyperparameters were chosen. The hyperparameter optimization system was developed using Python 3.9.5, TensorFlow 2.5 library, and Keras 2.5 application program interface. Suitable model hyperparameters were selected by training candidate RBFNNet models with the Adam optimization algorithm and randomly sampled hyperparameters. The random search is an efficient method widely used for hyperparameter optimization (Li *et al.*, 2021; Bergstra and Bengio, 2012). All experiments were conducted on an Intel Core i7-8550U CPU clocked at 1.80 GHz with 8.00 GB of RAM (Zacaria, 2023).

Prior to searching for the best hyperparameters, a search space for hyperparameters was established. In Table (1), the bounds of the hyperparameter search space for the PeerTrustRBFNNet are presented. To establish these boundaries, some preliminary experiments were conducted.

The search for the best hyperparameters involved assessing 162 PeerTrustRBFNNet models during 24 h. The faster hyperparameter optimization in the first experiment, compared to the second experiment, was due to the discovery of an early successful PeerTrustRBFNNet model. The RBFNNet model was evaluated by measuring the Mean Square Error (MSE) value on the test sets. A value closer to zero indicates improved model performance, as emphasized by Elzwayie *et al.* (2017).

Table 1: Hyperparameter search bounds for PeerTrustRBFNNet model

Hyperparameter	Minimum value	Maximum value
β (bias of output layer)	0.0	2.0
Hidden nodes	1	200
Time series sliding window size	1	15
Learning rate	10^{-6}, 10^{-5}, $10^{-}4$, 10^{-3}, 10^{-2}, or 10^{-1}	

In the next section, the second experiment will be discussed.

Experiment 2-ARDSTrustRBFNNet

Experiment 1 detailed the development of the RBFNNet trust model, named PeerTrustRBFNNet, utilizing the trust model introduced by Xiong and Liu (2004). The central emphasis was on validating the trust data generation model, PSTDG-PeerTrust, taking into account possible errors in the program's design, data handling, and trust quantification. Experiment 2 demonstrates the proposed solution by employing the Amazon Relational Database Service (ARDS) as an example, with a specific emphasis on the context of high availability and durability. This experiment also underscores the importance of acquiring expert knowledge and calculating trust values, addressing the challenge of lacking predefined algorithms or equations for the ARDS trust calculation.

PSTDG-ARDS Model Development

The initial four elements of the PSTDG framework, presented in the section titled "PSTDG Framework" and demonstrated in Experiment 1, will be utilized in developing the data generation model referred to as PSTDG-ARDS:

1) Identification of the Necessity for SDG
2) The necessity is explained in the section titled "Challenges in Collecting Trust Data"
3) Expert knowledge gathering

In this comprehensive process, all information, including qualitative expert insights on trust data, is gathered. According to the new definition of trust, trust calculation entails utilizing information from sources aligned with standards or guidelines, leading to the attainment of desired attributes for trust values. These properties depend on the data used or the applied standards or guidelines. Determining expert knowledge involves identifying input sources, entity features, critical factors for trust evaluation, and applicable standards. In the case of Experiment 1, an algorithm for this purpose was available, but the absence of one for the ARDS prompts an examination of its Service Level Agreement (SLA), published features, and standards to gather expert knowledge in the contexts of high availability and durability. The ARDS has specific features and standards, including “Automated Backups”, “Database Snapshots”, “Multi-AZ Deployments”, and “Quick Disaster Recovery” within the context of high availability and durability (Amazon RDS features, 2024). The ARDS specifies an SLA in terms of Monthly Uptime Percentage (MUP), serving as a standard for trust evaluation. In the calculation of trust value within the realm of high availability and durability, it is essential to consider both the SLA and MUP or the percentage availability of all features offered by the ARDS within this specific category.

Following an evaluation of the ARDS's SLA, MUP, or percentage availability, along with the examination of its features within the scope of high availability and durability and conventional trust calculation algorithms, a set of discernible parameters emerges. These parameters are used to compute trust within the ARDS’s high availability and durability context. This encompasses aspects such as the number of features provided in this specific context and the percentage availability or success of key features.

The parameters also include the total number of transactions conducted, the outcome of credit return success or failure in transactions, the cumulative count of credit returns that have failed to date, and trust values derived from prior transactions or performance.

The ARDS ensures an MUP of 99.95% (Amazon RDS Service Level Agreement, 2024; 2019). Standards for the ARDS are anchored in MUP, guiding credit back policies: A full refund is guaranteed for MUP values below 95, 25% if MUP is <99% and >= 95, and 10% if percentage availability is <99.95% and >= 99%. The trust value decreases with a decline in MUP or percentage availability. The trust value is reduced in case of a credit return failure, with the reduction amount increasing with each failure. A decrease in trust value is also associated with a decline in percentage availability.

A decrease in the ARDS's MUP, coupled with a failure to fulfill the offered credit return, results in a reduction in trust value. The distilled information from the above observations is as follows and might be regarded as expert knowledge for calculating ARDS trust values in the context of high availability and durability (Zacaria, 2023):

1. The ARDS's trust should be contingent on the MUP's success. It should exhibit a direct correlation with the MUP, and a reduction in trust value is necessary as the percentage availability diminishes
2. The trust value ought to be computed based on the entirety of the MUP, the percentage availability, or the percentage success of all features obtained from peers or customers
3. The trustworthiness assessment of the ARDS

concerning high availability and durability hinges on the outcome of credit return success or failure. The extent of the trust value decrement should escalate with each instance of credit return failure

The knowledge from the above-given expert knowledge will be utilized in formulating the constraints.

Compile constraints.

The ARDS's trust value can be calculated differently, depending on the context, and it was noted that no published model specifically addresses the ARDS's trust values. Drawing from expert knowledge, equations need to be developed for trust calculations, offering users flexibility in choosing the calculation method based on their needs or circumstances.

Two implementation approaches for generating a PSTDS are outlined. In Instance 1, termed the Basic Trust Matrix, the computation of the trust value involves averaging the satisfaction of each customer in the MUP received, specifically identified as the availability context factor. The *ARDS* trust score is determined by a formula known as the Basic Trust Matrix (Zacaria, 2023):

$$\mathrm{T}(ARDS) = \sum_{i=1}^{I(p)} S(P, i) \tag{3}$$

where, *S(P, i)* represents customer satisfaction on the MUP, ranging from 0 to 1, and $\mathrm{I}(P)$ represents the total count of feedback from peers (customers) within a recent time frame (Zacaria, 2023).

The $S(P, i)$ can either be a single value or a weighted average approach for each of the features offered by the ARDS in the specific context. The weighted average satisfaction is computed as below:

$$\mathrm{S}(P,i) = \gamma_1 \cdot S(p_1,i) + \gamma_2 \cdot S(p_2,i) + \gamma_3 \cdot S(p_3,i) + \gamma_4 \cdot S(p_4,i) \tag{4}$$

where, $\gamma_1, \ldots \gamma_4$ represent the weight factors corresponding to the availability of features, such as "Automated Backups", "Database Snapshots", "Multi-AZ Deployments", and "Quick Disaster Recovery", as outlined at the beginning of this section and $\mathrm{S}(p_1, i), \ldots, S(p_4, i)$ denote customer satisfaction scores (or peer/customer feedback) between 0 and 1 for each feature, as outlined above. In this instance, the first two pieces of expert knowledge identified in this experiment are used.

In Instance 2, the trust value of the ARDS is adjusted by incorporating both the availability context factor and the credit return success context factor. The modified equation is given below:

$$\mathrm{T}(ARDS) = \alpha \cdot \sum_{i=1}^{I(p)} S(P,i) - \beta \cdot CRF(ARDS) \tag{5}$$

where, $CRF(ARDS)$ is the credit return success context factor, deducted from the trust value. In addition, α and β are the weight factors for availability and the durability factor, respectively. These weights can be adjusted to control the reputation level reduction in case of a credit return failure from the *ARDS*. The $CRF(ARDS)$ value is determined by the ratio of credit failures to the required credit returns during a specified period:

$$\mathrm{C}RF(ARDS) = \frac{I(c_f)}{I(c_{rr})} \tag{6}$$

where, $I(\mathrm{c_f})$ represents the total credit failures and $I(c_{rr})$ is the total required credit returns within the specified period. By integrating the MUP satisfaction as a weighted average and credit return success as the ratio of failures to required returns during a given period, the ARDS trust value is derived as below:

$$\mathrm{T}(ARDS) = \alpha \cdot \sum_{i=1}^{I(p)} \gamma_1 \cdot S(p_1,i) + \gamma_2 \cdot S(p_2,i) + \gamma_3 \cdot S(p_3,i) + \gamma_4 \cdot S(p_4,i) - \beta \frac{I(c_f)}{I(c_{rr})} \tag{7}$$

The formulas and parameters provided offer a customizable approach to calculating the trust value of the ARDS, emphasizing its high availability and durability in different scenarios. The user's choices in weight factors and feature contributions directly influence the resulting trust values. In the second approach, the trust score of the ARDS is determined by using all three expert knowledge which will be used to develop the PSTDG-ARDS model.

Develop PSTDG-ARDS model.

The construction of the PSTDG-ARDS model is done using Instance 2 for the generation of a PSTDS, as previously stated. Two MS SQL stored procedures are used to develop the model. The first procedure stores FeatureCount, PercentageUptime, and MonthNames, along with their corresponding values, in the SQL database. The second procedure incorporates nuanced weight factors for features, availability, and credit return success, orchestrating monthly feedback submissions from peers and iteratively calculating updated trust values for the ARDS. Expert Knowledge 1 and 2 play pivotal roles in influencing the availability context, while Expert Knowledge 3 dynamically affects the credit return success context. The trust values undergo continual updates after each carefully managed feedback submission.

Validation of the PSTDG-ARDS

The PSTDG-ARDS incorporates all the identified expert knowledge in the computation of the trust score for the ARDS. Validation followed the steps outlined in the section titled "Sensitivity and What-if analyses" and demonstrated in Experiment 1. The what-if scenarios, aligned with expert knowledge, explore the impact of varying satisfaction levels, feedback from different sources, and credit return failures on the trust value of the ARDS. The validation process confirmed the PSTDG-

ARDS model's ability to generate a valid trust dataset for training the ARDSTrustRBFNNet model (Zacaria, 2023).

PSTDS Generation Using PSTDG-ARDS

The PSTDG-ARDS model produced 13727 data points with 13 inputs. The dataset produced by the PSTDG-ARDS model includes the count of transactions that have occurred thus far, the month's name, the name of the peer, the percentage of satisfaction received from peers or customers for each of the four features, and the overall satisfaction calculated, using four weight factors. Additionally, the model considers the qualification of the peer or customer for credit refunds from the ARDS, whether the ARDS has successfully returned the credit to eligible peers or customers, the overall count of credit return failures, the cumulative count of credit returns needed up to this point and the updated trust score for the ARDS following the transaction or submission of monthly feedback from each peer or customer (Zacaria, 2023).

Construction of ARDSTrustRBFNNet Model

Similar to the first experiment, the construction of the ARDSTrustRBFNNet model followed the same three steps which include the preparation of a dataset, selection of the best ARDSTrustRBFNNet model, and assessing the selected model. The search for the best hyperparameters involved assessing 4531 ARDSTrustRBFNNet models during 250 h. Experiment 2 utilized the identical search space bounds for hyperparameters as Experiment 1, which is given in Table (1). In the next section, the evaluation and results obtained from the two experiments will be discussed.

Results and Discussion

During the initial trial, the primary emphasis was placed on validating the PSTDG-PeerTrust model, which was used to generate a PSTDS for the PeerTrustRBFNNet model. Li and Ling's PeerTrust model offered the theoretical trust calculation equation, but the development process of the PSTDG-PeerTrust model did not necessarily have to produce a valid trust dataset. Therefore, SWA was used to validate the model based on the new definition. Developing a large trust dataset was the major challenge in training an RBFNNet, which was resolved by the PSTDG-PeerTrust model. Table (2) shows the hyperparameters of the best PeerTrustRBFNNet model found, which was trained for 100 epochs using the Adam optimization algorithm and had an MSE of $2.58 \cdot 10^{-4}$.

Table 2: Hyperparameters of the best PeerTrustRBFNNet model

Hyperparameter	Value
β	$7.13 \cdot 10^{-2}$
Hidden nodes	72
Sliding window size	1
Learning rate	10^{-3}

Table 3: Hyperparameters of the best ARDSTrustRBFNNet model

Hyperparameter	Value
β	$7.71 \cdot 10^{-1}$
Hidden nodes	126
Sliding window size	14
Learning rate	10^{-4}

In the second trial, the PSTDG-ARDS model was formulated, serving as a dedicated model for generating a PSTDS employing the ARDS (Zacaria, 2023). This model was utilized to demonstrate the applicability of the proposed solution to real-world challenges. The main focus was on developing and validating a model for the ARDS, as there were no existing algorithms or equations to perform a trust calculation. Therefore, it became essential to demonstrate the acquisition of expert knowledge and the quantification of trust in the absence of any established methodology. The PSTDG-ARDS model was validated to ensure it satisfies expert knowledge. The validity of the PSTDS was established through the validation of the PSTDG-ARDS model. The model underwent verification as described in the section titled “Sensitivity and What-if analyses”. This model was developed to address the issue of insufficient data for training an RBFNNet model to compute trust scores for the ARDS. After constructing the PSTDS, the best RBFNNet model was determined. The hyperparameter values are listed in Table (3). This model was trained for 128 epochs and resulted in an MSE value of $7.72 \cdot 10^{-6}$.

The paper will conclude in the following section, offering concluding remarks.

Conclusion

Quantifying trust between electronic entities is one of the challenging issues in a digital environment, as trust is a generic concept. This study suggests a new method for trust calculation using an RBFNNet. It addresses issues such as heightened complexity in calculations, time, storage of data, and accessibility to extensive datasets for trust calculation. In the domain of computing and technology, a universally accepted definition for the term trust has not been established. Therefore, a new definition has been proposed that defines trust as having three dimensions: Context, calculation or dimension of quantification of trust, and sources of information utilized for the computation. The PSTDG framework can be used to create a legitimate and entirely synthetic trust dataset, solving the scarcity of trust data. A novel definition was crafted for authenticating synthetic trust datasets. The verification of the model for generating trust data was illustrated through a three-step process that included the validation of two models: PSTDG-PeerTrust and PSTDG-ARDS. The proposed experimental design process consisting of four steps was successfully demonstrated through two experiments to build an RBFNNet model for

determining trust values between electronic entities. Calculating trust values no longer requires large data storage and access by an RBFNNet, thus reducing calculation time. Employing an RBFNNet to calculate trust would lead to a decrease in the complexity of the calculations. When using an RBFNNet, the complexity of the network will have no impact on the trust calculation complexity. Using an RBFNNet to calculate trust and an increased number of parameters to quantify trust will also avoid adding complexity to the trust calculation. Finally, this study demonstrates the successful development of a trust calculation method using an RBFNNet.

This study introduced an RBFNNet for trust calculation, utilizing a dataset generated from a validated PSTDG model. However, performance and accuracy comparisons with other neural network models were not conducted. Future research could focus on comparative studies and exploring different neural network architectures to improve performance and predict future trust values. Additionally, the failure to accurately gather all the required expert knowledge can affect data generation, which is a limitation associated with the synthetic data generation process.

Acknowledgment

We gratefully acknowledge the funding received from the Unit for Data Science and Computing (UDSC) at the North-West University in South Africa and the access to their equipment, which was instrumental in facilitating the experimentation and analysis conducted in this study.

Funding Information

The UDSC provided funding for this study.

Author's Contributions

Smitha Zacaria: Research design, experimentation, and writing of the manuscript.

Tiny Du Toit: Research and critical reviewing of the manuscript.

Ethics

This article is original work and has not been submitted for publication elsewhere. The authors have read and approved the manuscript.

References

Aberer, K., & Despotovic, Z. (2001). Managing Trust in a Peer-2-Peer Information System. *Proceedings of the Tenth International Conference on Information and Knowledge Management*, 310–317. https://doi.org/10.1145/502585.502638

Al-Shargabi, B. (2016). Security Engineering for E-Government Web Services: A Trust Model. *2016 International Conference on Information Systems Engineering (ICISE)*, 8–11. https://doi.org/10.1109/icise.2016.17

Alawneh, M., & Abbadi, I. M. (2022). Integrating Trusted Computing Mechanisms with Trust Models to Achieve Zero Trust Principles. *2022 9th International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, 1–6. https://doi.org/10.1109/iotsms58070.2022.10062269

Amazon RDS Service Level Agreement. (2019). Amazon. https://aws.amazon.com/rds/sla/

Amazon RDS features. (2024). Amazon. https://aws.amazon.com/rds/features/

Amazon RDS Service Level Agreement. (2024). Amazon. https://aws.amazon.com/rds/sla/

Anderson, J. W., Kennedy, K. E., Ngo, L. B., Luckow, A., & Apon, A. W. (2014). Synthetic Data Generation for the Internet of Things. *2014 IEEE International Conference on Big Data (Big Data)*, 171–176. https://doi.org/10.1109/bigdata.2014.7004228

Au, R., Looi, M., & Ashley, P. (2001). Automated Cross-Organisational Trust Establishment on Extranets. *Proceedings Workshop on Information Technology for Virtual Enterprises. ITVE 2001*, 3–11. https://doi.org/10.1109/itve.2001.904483

Barse, E. L., Kvarnstrom, H., & Jonsson, E. (2003). Synthesizing Test Data for Fraud Detection Systems. *19th Annual Computer Security Applications Conference, 2003. Proceedings.*, 384–394. https://doi.org/10.1109/csac.2003.1254343

Basile, F., & Ferrara, L. (2023). Validation of Industrial Automation Systems Using a Timed Model of System Requirements. *IEEE Transactions on Control Systems Technology*, *31*(1), 130–143. https://doi.org/10.1109/tcst.2022.3173890

Bergstra, J., & Yoshua, B. (2012). Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, *13*(10), 281–305.

Bhaumik, P., Sayeem Reaz, A., Murayama, D., Suzuki, K.-I., Yoshimoto, N., Kramer, G., & Mukherjee, B. (2015). IPTV Over EPON: Synthetic Traffic Generation and Performance Evaluation. *Optical Switching and Networking*, *18*, 180–190. https://doi.org/10.1016/j.osn.2014.05.007

Broomhead, D. S., & Lowe, D. (1988). *Radial Basis Functions, Multi-Variable Functional Interpolation and Adaptive Networks*. Royal Signals and Radar Establishment. https://apps.dtic.mil/sti/tr/pdf/ADA196234.pdf

Carley, K. M. (1996). Validating Computational Models. *Center for the Computational Analysis of Social and Organizational Systems CASOS Technical Report*, 1–24.

Chan, Y.-H., Correa, C. D., & Ma, K.-L. (2010). Flow-Based Scatterplots for Sensitivity Analysis. *2010 IEEE Symposium on Visual Analytics Science and Technology*, 43–50. https://doi.org/10.1109/vast.2010.5652460

Chien-Cheng, L., Pau-Choo, C., Jea-Rong, T., & Chein-I, C. (1999). Robust Radial Basis Function Neural Networks. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, *29*(6), 674–685. https://doi.org/10.1109/3477.809023

Christopher Frey, H., & Patil, S. R. (2002). Identification and Review of Sensitivity Analysis Methods. *Risk Analysis*, *22*(3), 553–578. https://doi.org/10.1111/0272-4332.00039

Das, P., & Debnath, S. (2020). A Trust Computing Model for Future Generation Networks. *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 1–4. https://doi.org/10.1109/icccnt49239.2020.9225462

Denning, D. (1993). A New Paradigm for Trusted Systems. *Proceedings on the 1992-1993 Workshop on New Security Paradigms*, 36–41. https://doi.org/10.1145/283751.283772

Elzwayie, A., El-shafie, A., Yaseen, Z. M., Afan, H. A., & Allawi, M. F. (2017). RBFNN-Based Model for Heavy Metal Prediction for Different Climatic and Pollution Conditions. *Neural Computing and Applications*, *28*(8), 1991–2003. https://doi.org/10.1007/s00521-015-2174-7

Gambetta, D. (2000). Can We Trust Trust? In *InTrust: Making and Breaking Cooperative Relations* (pp. 213–237). Blackwell: Department of Sociology.

Gitzel, R., Kotriwala, A., Fechner, T., & Schiefer, M. (2021). Using Synthetic Data to Train a Visual Condition Monitoring System for Leak Detection. *2021 IEEE Seventh International Conference on Big Data Computing Service and Applications (BigDataService)*, 196–200. https://doi.org/10.1109/bigdataservice52369.2021.00031

Grandison, T., & Sloman, M. (2003). Specifying and Analysing Trust for Internet Applications. In J. L. Monteiro, P. M. C. Swatman, & L. V. Tavares (Eds.), *Towards the Knowledge Society* (Vol. 105, pp. 145–157). Springer US. https://doi.org/10.1007/978-0-387-35617-4_10

Hang, C.-W., Kalia, A. K., & Singh, M. P. (2012). Behind the Curtain: Service Selection Via Trust in Composite Services. *2012 IEEE 19th International Conference on Web Services*, 9–16. https://doi.org/10.1109/icws.2012.96

Jones, W. W., Peacock, R. D., Forney, G. P., & Reneke, P. A. (2004). *Verification and validation of CFAST, A Model of Fire Growth and Smoke Spread*. National Institute of Standards and Technology. https://doi.org/10.6028/nist.ir.7080

Kleijnen, J. P. C. (1997). Sensitivity Analysis and Related Analyses: A Review of Some Statistical Techniques. *Journal of Statistical Computation and Simulation*, *57*(1–4), 111–142. https://doi.org/10.1080/00949659708811805

Kleijnen, J. P. C. (1998). *Validation of Simulation, with and without Real Data*. *22*, 21.

Koval, S. I. (2018). Data Preparation for Neural Network Data Analysis. *2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, 898–901. https://doi.org/10.1109/eiconrus.2018.8317233

Kuźniar, K., & Zając, M. (2017). Some Methods of Pre-Processing Input Data for Neural Networks. *Computer Assisted Methods in Engineering and Science*, *22*(2), 141–151.

Law, A. M. (2019). How to Build Valid and Credible Simulation Models. *2019 Winter Simulation Conference (WSC)*, 8–11. https://doi.org/10.1109/wsc40007.2019.9004789

Li, W., Y. Ng, W. W., Wang, T., Pelillo, M., & Kwong, S. (2021). HELP: An LSTM-Based Approach to Hyperparameter Exploration in Neural Network Learning. *Neurocomputing*, *442*, 161–172. https://doi.org/10.1016/j.neucom.2020.12.133

Li, X., & Liu, L. (2002). Building Trust in Decentralized Peer-to-Peer Electronic Communities. *Fifth International Conference on Electronic Commerce Research (ICECR-5)*, 1–15.

Lin, G.-F., & Wu, M.-C. (2011). An RBF Network with a Two-Step Learning Algorithm for Developing a Reservoir Inflow Forecasting Model. *Journal of Hydrology*, *405*(3–4), 439–450. https://doi.org/10.1016/j.jhydrol.2011.05.042

Lu, Z., Delaney, D. T., & Lillis, D. (2023). A Survey on Microservices Trust Models for Open Systems. *IEEE Access*, *11*, 28840–28855. https://doi.org/10.1109/access.2023.3260147

Nawi, N. M., Atomi, W. H., & Rehman, M. Z. (2013). The Effect of Data Pre-processing on Optimized Training of Artificial Neural Networks. *Procedia Technology*, *11*, 32–39. https://doi.org/10.1016/j.protcy.2013.12.159

Mohsenzadeh, A., & Motameni, H. (2015). A Trust Model Between Cloud Entities Using Fuzzy Mathematics. *Journal of Intelligent and Fuzzy Systems*, *29*(5), 1795–1803. https://doi.org/10.3233/ifs-151657

Moody, J., & Darken, C. J. (1989). Fast Learning in Networks of Locally-Tuned Processing Units. *Neural Computation*, *1*(2), 281–294. https://doi.org/10.1162/neco.1989.1.2.281

Mui, L., Mohtashemi, M., & Halberstadt, A. (2002). A Computational Model of Trust and Reputation. *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, 2431–2439. https://doi.org/10.1109/hicss.2002.994181

Nguyen, Q. V. H., Zheng, K., Weidlich, M., Zheng, B., Yin, H., Nguyen, T. T., & Stantic, B. (2018). What-If Analysis with Conflicting Goals: Recommending Data Ranges for Exploration. *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, 89–100. https://doi.org/10.1109/icde.2018.00018

Olmedilla, D., Rana, O. F., Matthews, B., & Nejdl, W. (2006). Security and trust issues in semantic grids. *Dagstuhl Seminar Proceedings*, 1–11. https://doi.org/10.4230/DagSemProc.05271.11

Pillai, G. G., Putrus, G. A., & Pearsall, N. M. (2014). Generation of Synthetic Benchmark Electrical Load Profiles Using Publicly Available Load and Weather Data. *International Journal of Electrical Power and Energy Systems*, *61*, 1–10. https://doi.org/10.1016/j.ijepes.2014.03.005

Ranja, F., Nababan, E. B., & Candra, A. (2023). Synthetic Data Generation Using Time-Generative Adversarial Network (Time-GAN) to Predict Cash ATM. *2023 International Conference on Computer, Control, Informatics and Its Applications (IC3INA)*, 418–423. https://doi.org/10.1109/ic3ina60834.2023.10285809

Ruslan, F. A., Samad, A. M., Zain, Z. M., & Adnan, R. (2013). Modelling flood Prediction Using Radial Basis Function Neural Network (RBFNN) and Inverse Model: A Comparative Study. *2013 IEEE International Conference on Control System, Computing and Engineering*, 577–581. https://doi.org/10.1109/iccsce.2013.6720031

Rykiel, E. J. (1996). Testing Ecological Models: The Meaning of Validation. *Ecological Modelling*, *90*(3), 229–244. https://doi.org/10.1016/0304-3800(95)00152-2

Sagar, S., Mahmood, A., Sheng, Q. Z., Zhang, W. E., Zhang, Y., & Pabani, J. K. (2024). Understanding the Trustworthiness Management in the Social Internet of Things: A Survey. *Computer Networks*, *251*, 110611. https://doi.org/10.1016/j.comnet.2024.110611

Sargent, R. G. (2020). Verification and Validation of Simulation Models: An Advanced Tutorial. *2020 Winter Simulation Conference (WSC)*, 16–29. https://doi.org/10.1109/wsc48552.2020.9384052

Scheidegger, A., & Maurer, M. (2012). Identifying Biases in Deterioration Models Using Synthetic Sewer Data. *Water Science and Technology*, *66*(11), 2363–2369. https://doi.org/10.2166/wst.2012.471

Shakya, S., Yuan, H., Chen, X., & Song, L. (2011). Application of Radial Basis Function Neural Network for Fishery Forecasting. *2011 IEEE International Conference on Computer Science and Automation Engineering*, 287–291. https://doi.org/10.1109/csae.2011.5952682

Singal, H., & Kohli, S. (2016). Trust Necessitated through Metrics: Estimating the Trustworthiness of Websites. *Procedia Computer Science*, *85*, 133–140. https://doi.org/10.1016/j.procs.2016.05.199

Tahta, U. E., Sen, S., & Can, A. B. (2015). GenTrust: A Genetic Trust Management Model for Peer-to-Peer Systems. *Applied Soft Computing*, *34*, 693–704. https://doi.org/10.1016/j.asoc.2015.04.053

Thacker, B. H., Doebling, S. W., Hemez, F. M., Anderson, M. C., Pepin, J. E., & Rodriguez, E. A. (2004). *Concepts of Model Verification and Validation*. Los Alamos National La. http://inis.iaea.org/search/search.aspx?orig_q=RN:36030870

Thorve, S., Vullikanti, A., Mortveit, H. S., Swarup, S., & Marathe, M. V. (2022). Fidelity and Diversity Metrics for Validating Hierarchical Synthetic Data: Application to Residential Energy Demand. *2022 IEEE International Conference on Big Data (Big Data)*, 1377–1382. https://doi.org/10.1109/bigdata55660.2022.10020837

Tsvetovat, M., & Carley, K. M. (2016). Generation of Realistic Social Network Datasets for Testing of Analysis and Simulation Tools. *SSRN*, 1–27. https://doi.org/10.2139/ssrn.2729296

W3C. (2002). *Web Standards*. https://www.w3.org/standards/xml/security

Wang, C., & Chi, C. (2006). Quantitative Trust Based on Actions. *2006 IEEE International Conference on Web Services (ICWS'06)*, 178–188. https://doi.org/10.1109/icws.2006.109

Wang, J., Jing, X., Yan, Z., Fu, Y., Pedrycz, W., & Yang, L. T. (2021). A Survey on Trust Evaluation Based on Machine Learning. *ACM Computing Surveys*, *53*(5), 1–36. https://doi.org/10.1145/3408292

Wang, S., Zhang, L., & Wang, S. (2008). A Quantitative Evaluation Approach of Subjective Trust for E-Commerce. *2008 International Conference on Computational Intelligence for Modelling Control and Automation*, 761–766. https://doi.org/10.1109/cimca.2008.8

Wang, Y., & Vassileva, J. (2003). Trust and Reputation Model in Peer-to-Peer Networks. *Proceedings Third International Conference on Peer-to-Peer Computing (P2P2003)*, 150–157. https://doi.org/10.1109/ptp.2003.1231515

Witkowski, M., & Pitt, J. (2000). Objective Trust-Based Agents: Trust and Trustworthiness in a Multi-Agent Trading Society. *Proceedings Fourth International Conference on MultiAgent Systems*, 463–464. https://doi.org/10.1109/icmas.2000.858526

Xiong, L., & Liu, L. (2003). A Reputation-Based Trust Model for Peer-to-Peer E-Commerce Communities. *EEE International Conference on E-Commerce, 2003. CEC 2003*, 275–284. https://doi.org/10.1109/coec.2003.1210262

Xiong, L., & Liu, L. (2004). PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities. *IEEE Transactions on Knowledge and Data Engineering*, *16*(7), 843–857. https://doi.org/10.1109/tkde.2004.1318566

Yong-Sheng, Z., & Ying, W. (2010). Research on Trust-Authorization-Based Access Control Model for Web Services. *2010 Second International Conference on Networks Security, Wireless Communications and Trusted Computing*, 454–457. https://doi.org/10.1109/nswctc.2010.113

Zacaria, S. (2023). *An Alternative Trust Calculation Method Using Radial Basis Function Neural Networks*. North-West University.