Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2021 SCPE. Volume 22, Issues 4, pp. 425–444, DOI 10.12694/scpe.v22i4.1952

TRUST MANAGEMENT IN THE WORLD OF CLOUD COMPUTING. PAST TRENDS AND SOME NEW DIRECTIONS

MAHREEN SALEEM, M.R WARSI, SAIFUL ISLAM, AREESHA ANJUM AND NADIA SIDDIQUII [*]

Abstract. Over the past years, Cloud computing has become one of the most influential information technologies to combat computing needs because of its unprecedented advantages. In spite of all the social and economic benefits it provides, it has its own fair share of issues. These include privacy, security, virtualization, storage, and trust. The underlying issues of privacy, security, and trust are the major barriers to the adoption of cloud by individuals and organizations as a whole. Trust has been the least looked into since it includes both subjective and objective characteristics. There is a lack of review on trust models in this research domain. This paper focuses on getting insight into the nomenclature of trust, its classifications, trust dimensions and throws an insight into various trust models that exist in the current knowledge stack. Also, various trust evaluation measures are highlighted in this work. We also draw a comparative analysis of various trust evaluation models and metrics to better understand the notion of trust in cloud environments. Furthermore, this work brings into light some of the gaps and areas that need to be tackled toward solving the trust issues in cloud environments so as to provide a trustworthy cloud ecosystem. Lastly, we proposed a Machine Learning backed Rich model based solution for trust verification in Cloud Computing. We proposed an approach for verifying whether the right software is running for the correct services in a trusted manner by analyzing features generated from the output cloud processed data. The proposed scheme can be utilized for verifying the cloud trust in delivering services as expected that can be perceived as an initiative towards trust evaluation in cloud services employing Machine learning techniques. The experimental results prove that the proposed method verifies the service utilized with an accuracy of 99%.

Key words: Cloud SLA, Trust, Trust Evaluation, Trust Management, SRM, Ensemble classifier

AMS subject classifications. 65M25, 68M14

1. Introduction.

Cloud Computing as a paradigm shift, has enabled convenient and on-demand /immediate accessibility to the available pool of shared resources. With the unstoppable advancements in the networking technologies and rapidly increasing demand for computing resources, cloud adoption has become as inevitable and convenient as the daily-life utilities like gas, water, and electricity. Cloud service providers (CSP) commonly utilize Virtual Machine (VM) [1] technologies backed up by the contemporary data centers to dynamically provide computing services like ubiquitous network access, computing-on-demand, rapid resource elasticity on a pay-as-you-use basis [2]. The different types of services that Cloud computing offers consist of 'Infrastructure as a Service (IaaS)', 'Platform as a Service (PaaS)', and 'Software as a Service (SaaS)'. With IaaS, a CSP offers computing, storage or networking infrastructure to its customers for use. With Platform as a Service (PaaS), a CSP allows customers to leverage the shared location-independent cloud resources from the resource pool of CSP to operate custom applications. Software as a Service (SaaS), allows consumers to employ softwares running on CSP's infrastructure. For most customers, the facility of pay-as-you-go is a great motivation to migrate to Cloud, as it relieves the user of the planning and maintenance of the underlying architecture.

Yet, in spite of the swift acceleration of cloud services adoption, most IT administratives still are skeptical to commend a "cloud-first" approach. Even worse, some desist to utilize any of the cloud services, quoting privacy and security issues, functional challenges and most prominently the diminishing of control over the data after it goes beyond the boundary [3]. Undoubtedly, the issue of trust management is one of the most complex in cloud computing systems where a collection of services, applications, and nodes function collectively to serve each other [4] [5] [6]. This calls for the deployment of urgent means that uphold awareness for transparency, accountability, and, governance of the CSPs. To improve adoption of cloud, trust is a must desired criteria that

[*] Department of Computer Engineering, Aligarh Muslim University, India (mehkhan27@gmail.com, warsimr@yahoo.com, saifulislam@zhcet.ac.in, aressha.anjum@zhcet.ac.in, nadiasid2134@gmail.com)

if guaranteed, would motivate individuals and organizations at whole to migrate to the cloud [7] [8]. The solution is to build a holistic cloud trust scheme — one that involves potential business and IT stakeholders to deliver accurate checks and balances to create secure cloud environment that allows a controlled and cost-effective cloud investment.

By building a cloud trust framework to evaluate and monitor, upgrade and enhance their cloud environment, and to certify and comply with it, IT experts can transform cloud fear into an possibility to tackle increasingly complicated security and privacy issues [3] [9]. However, despite of the significance of cloud trust evaluation in trust management, there is a lack of relevant literature addressing this challenge comprehensively. Considering trust to play a pivot role in Cloud systems the main objective of this paper is to survey the existent trust models and to highlight the significant contemporary trust management challenges faced in Cloud Computing and finally to propose a solution for verifying trust in Cloud service utilization.

Briefly, the outlines of this paper are as follows:

- Providing the basic semantics, taxonomy, and notion of trust specific to Cloud Computing.
- Offering the overview of Trust evaluation mechanism in Cloud environment.
- Systematically discussing the significant Trust Models in Cloud Computing.
- Comparing the reviewed Trust models and mechanisms, and outlining their major features.
- Highlighting the open issues and suggestions to establish the trust in cloud computing systems.
- Proposing a Machine Learning backed Rich model based solution for trust verification in Cloud Computing.

The rest of this paper is structured as follows. Section 2 introduces the trust semantics and terminologies. Section 3 offers the overview of trust in Cloud context and gives insight into challenges in trust management. Section 4 describes various trust dimensions and trust evaluation measures in cloud systems. Section 5 presents an overview of a few proposed Trust cloud models in the literature. Section 6 presents the discussion and outlines some open issues. In section 7 we propose a Machine Learning backed Rich model based solution for trust verification in Cloud Computing. Finally, section 8 ends this paper.

2. Semantics Of Trust. Trust is a complex notion; therefore a multidisciplinary approach is required to describe it. Trust is, for the most part, characterized as “the certainty levels in something or somebody” [10]. In IT environments, this implies that your counterpart functions in accordance with the defined protocols. Trust is to be expanded by alleviating technical and psychological boundaries to utilizing cloud services.

According to Leonard.G et al. [11], trust in something can be described as the level of certainty that an object will perform in an acceptable fashion. This assurance can be related to the ‘quality of service’ or the ‘security and privacy policies’ that the object follows. Trust management in the system can assist in establishing collaborations among new devices that have joined the network. Nodes may be reported as safe or unsafe, subject to the trust level [12]. Rousseau and colleagues [13], describe trust as a psychological state of accepting to being intentionally vulnerable on the basis of positive expectations of someone else’s behavior. In computer science, trust is considered to be quantifiable, and object X’s trust in object Y for any service S implies that X believes that the behavior of Y will be satisfactory for a specified time period within a defined context for the service S.

Reputation and trust are recurrently used exchangeably; though the conception of both is quite very much alike but not the same [11]. Reputation is the belief someone has regarding something. Trust information can be collected from the reputation; however trust formation takes other components as well. Reputation is established on the basis of past behavior of an entity while trust predicts its future behavior [14].

Often times, security is associated with trust. Security guards systems against the malicious entities. Nonetheless, there is a need of essential pre-configured and established information well equipped to protect network entities. In trust context, security measures could be viewed as transporters of trust from the source where it is established to the point where it is required [15]. If the two interacting entities share their key, they have officially established trust with one another.

2.1. Trust Terminology. Alhanahnah et al. [16] presented a taxonomy of trust factors in their article and they define trust-related terms to avoid ambiguity as they often have varying connotations in the literature. Those are defined as follows:

Trust factors (TFs): Trust factors are the criteria that are considered for trust evaluation in CSPs. Security, privacy, and data management are some instances of high-level TFs.

Trust indicators (TIs): These are the indicators that represent trust factors. Score (e.g. 90 percent), or rating (4 out of 5 stars) are TIs that represent the trust factor reputation.

Trustor: An agent that trusts some other entity is termed as trustor. In cloud context, a trustor corresponds to a consumer or user.

Trustee: An entity trusted by the trustor is the trustee. In cloud context, a CSP corresponds to the trustee entity. Two entities are involved in building trust –trustor and trustee – both have two different goals. The essential role of the trustor is to assess the accurate trustworthiness, that is to predict the future behavior, of the trustee. However, the goal of the trustee is to acquire the creditably best trust values.

Trustworthiness: Trustworthiness is perceived as the collection of all significant trust factors.

Trust decision: A trustor decides to connect with the trustee only if it's trust value is perceived to be sufficiently high at that instant. This decision is termed as the trust decision.

3. Understanding Trust in Cloud Context. First off, we sketch the perception of trust in cloud computing and feature notable concerns that highlight the need for constructing and managing the trust in cloud context. Then we put forward the taxonomy and outline how it can be enforced to practical scenarios.

Trust is widely described in generic literature, however, it is still in its infancy in the cloud computing setting. In IT environments, the notion of trust pertains to further than security. It encompasses reliability, dependability, integrity, and capability to execute a service. Trust additionally propels collaboration among groups. In online communities having a majority of unknowns, the trust metric would predict the degree to which a user can trust another user in the network [17]. Trust in cloud computing setting has three important referents: trust in the Cloud provider; trust in the Cloud services that CSP has to offer; and trust in the Cloud as a technology itself. Among the three potential referents the most difficult one seems the trust in the Cloud itself as it involves motivating users to adopt cloud computing as a technology.

Fujitsu Research Institute conducted a 2010 survey [18] that derived 88% of potential cloud end users are concerned mainly about who else has access to their information and necessitated more transparency of what happens in the physical servers at the backend. Surveys suchlike demonstrate the pressing need for IT executives and researchers to address the obstacles to trust urgently. While there exist preventive measures like encryption, ID profiling based access control etc. to mitigate privacy and security risks, they are not sufficient. Due to insufficient confidences in the Cloud, many users are reluctant to migrate their data to the Cloud or to utilize the shared computing resources provided by it. Security, privacy and trust issues have become one of the major barriers to the adoption of Cloud services by many individuals and organizations. Cloud Service providers do not provide guaranteed service performance. In terms of dependability, some initial approaches exist in levels of service availability and reliability [19] [20] [21].

Due to the very complex and distributed nature of cloud computing, cloud consumers lose control over their information, as the data resides on the distributed cloud data centers located across different geographical locations. Concerns over the security of remotely stored information and the consequent ownership and diminishing transparency issues are aggravated by the fact that most customers may not be conscious of the underlying security and privacy measures enforced by the CSP [22]. Building trust plays a significant role to combat these challenges [1] [6] [16] [23]. Undeniably, trust can remunerate the absence of control and build the confidence of the consumers in the security and privacy measures executed by the CSP. As such, a CSP 's trustworthiness is a denotation that the CSP conforms to the security and privacy benchmarks, and meets the safety prerequisites of its consumers [16].

Challenges of Trust Management in Cloud Computing. We discuss in what follows the trust management challenges associated with establishing trust among consumers and CSPs. It categorizes trust management challenges into various levels based on different approaches as outlined in this section.

The Cloud Security Alliance listed some top threats in cloud computing in their report 'Top Threats to Cloud Computing V1.0' [22]. There are many challenges in Cloud Computing in addition to internal and external threats, software bugs, hardware failures, server misconfigurations, etc. [24] [25] [26]; the most significant ones are being highlighted below:

- Security and Privacy
- Integrity
- Interoperability and Portability
- Reputation
- Virtualization
- SLA
- Standardization
- Query and access
- Service quality, and
- Trust management

Trust management facilitates the establishment of trust between entities and specifies the structure and mechanisms that enable the trust to be manifested in a system. For the first time, trust management conception was introduced by 'M. Blaze, J. Feigenbaum, and J. Lacy' [27]. They outline the definition of trust management as "the problem of figuring based on formulated security policies and security credentials if a set of security credentials of an entity satisfies the security policies". To guarantee the trustworthiness of an entity, trust management describes the two aspects of the system trust: 1) What information to gather, and 2) How to gather that information. Trust management encompasses three major components: i) Trust establishment ii) Trust update and iii) Trust revoke [12].

In service oriented IoT, the goal of trust management is to guarantee whether connecting to the service provider is safe in terms of reliability, security, and availability [28] [29] [30]. This trust-related information is either stored centrally or in a distributed setup prior to being delivered to the network.

Trust management of cloud services is a challenging issue. This is accredited to the unique characteristics of cloud services wherein millions of nodes, services, and applications are deployed to serve each other under a single umbrella. The nature of cloud computing is very dynamic where newer cloud services and nodes keep joining the network. Moreover, it is difficult to access the legitimacy of the user trust feedbacks [31]. Also, it is challenging to assign credibility to experienced users and track bad-mouthing in order to identify malicious trust feedback [32].

Navimpour et al. [33] in their survey paper categorized trust issues of cloud systems into four sub-classes, that comprise:

(a) How to define and access trust related to dynamic cloud systems.
(b) How to handle recommended trust information from the malicious entities.
(c) How to provide varying levels of service in accordance to the expected trust degree, and
(d) How to monitor trust values as they update with variation in time and context, and to update the trust information and adapt the system to the dynamic changes as and when they occur in time.

Practically, autonomic trust management is difficult to realize due to the ever-expanding scale of deployment of the cloud of things and the very dynamic nature of the cloud systems further complicates the task [2]. Ryan Ko et al. [10] mentioned the primary issues and challenges in building a trusted cloud environment through the establishment of detective controls and presented the 'TrustCloud' framework, that attends to the accountability issues in cloud computing through technical and policy-based techniques. The authors quote that in spite of auditability contributing a significant role in building trust, the present day leading CSPs (e.g. Amazon EC2/S3, Microsoft Azure etc.) still do not grant complete transparency and facilities to track and audit record access history and the provenience of the server (physical and virtual) usage [18]. At present, clients can at best have transparency to track performance metrics of the virtual hardware and monitor service event logs [34] [35].

4. Trust Management Overview.

4.1. Trust Dimensions. Trust computing is a subcategory of trust management that describes how trust information is collected, what trust features are used, and how the gathered trust values are aggregated to generate the final concluding trust values that are again broadcasted over the network. J. Guo et al. [36] classified trust computing methods in service-oriented IoT into five dimensions: i) Trust composition ii) Trust propagation iii) Trust update iv) Trust formation v) Trust aggregating.

Trust Composition: Trust composition is a major component of trust management and it defines the components that are contemplated in the computation of trust. The set of trust components can be categorized into two components: i) Quality of Service (QoS) component and ii) Social trust component. QoS trust is defined by the level of assurance in the node to deliver the requested service [37]. Social trust component is reflected from social affiliations of entities, their social networking etc and it could be employed to contribute to overall trust derivation.

Trust Propagation: This dimension of trust computation represents how to store the composed trust values in the network. In the literature, two major schemes are outlined; centralized and distributed. In the centralized trust propagation scheme a centralized entity, like the cloud, is responsible for acting as a datastore of trust information for all network entities. A centralized datastore of trust information makes message propagation simple. Also, the centralized server can be responsible for the processing of trust values. Every entity in the network has access to the same central information; hence limited storage issue in the resource-constrained entities is resolved. However, this causes the central node to eventually be a single point of breakdown. In a distributed approach, instead of a central entity, every object in the network has the obligation to compute and store the required trust values, and render recommendations to other objects. However, the distributed approach can suffer from attacks such as bad-mouthing.

Trust Update: Trust information once collected and propagated either centrally or in a distributed manner needs to be updated. When these trust values are updated, is described by the trust update dimension. Trust update can be either event-driven or time-driven. In the event-driven approach, trust update takes place after the occurrence of an event e.g. after completion of a transaction. In the time-driven approach, there is a periodic update in trust values at regular time intervals. In a distributed scenario time-driven approach is considered to be more suitable by virtue of it being energy persevering in contrast to an event-driven scheme [36]. Though, over the course of time the trust value of an entity may waver; hence, making the time-driven approach questionable and less accurate. To maximize accuracy and minimize energy consumption, an optimal time interval needs to be estimated.

Trust formation: Once the trust information is collected from multiple features, trust properties need to be weighted corresponding to their relevance. Thus, trust formation specifies how trust constitution takes place from composite trust features. Trust formation can take place either by single-trust, wherein one single trust property like QoS is considered, or by multi-trust where multiple properties are considered.

Trust aggregation: Once the trust attributes are finalized, trust information is to be collected from the network entities. Trust aggregation defines how these trust feature values are collected from other members in the network. In the current literature, the most common method for combining these trust experiences into a single value is the weighted sum method [36] [38]. Other techniques for aggregating the trust values include Bayesian inference, belief theory, fuzzy logic, and regression analysis [36]. Subjective logic, a type of probabilistic logic, is the most appropriate aggregation technique for fog computing. It takes source trust and its uncertainty into account.

4.2. Components of Trust Evaluation in Cloud Computing. To build confidences in cloud by mitigating challenges in cloud trust management, we primarily require the understanding of the key components impacting cloud trust. The criteria that are considered while evaluating cloud trust are termed as Trust factors (TFs) [16] . Security, privacy, and data management are the instances of high-level (in terms of significance) trust factors. Ryan KO et al. [10] specify the following significant components that affect trust viz:

1. Security - Techniques like encryption that makes it challenging for an unauthorized individual to gain access over some information [39].
2. Privacy – Protecting confidential data from exposure or leakage.
3. Accountability – Obligation of an individual or organization to be answerable and liable for delivering agreed-upon services.
4. Auditability – The relative simplicity of inspecting a framework or a domain. Poor auditability implies inadequately-maintained (or non-existent) evidences or records that empower proficient reviewing of procedures inside the cloud.

In the paper [40] M. Alhamad et al., developed a model for trust evaluation based on the five most significant components of cloud trust; which include security, availability, scalability, and usability parameters. The authors developed a trust evaluation scheme, for IaaS model with an e-learning application as a case study, using fuzzy-set theory since each of these trust properties is characterized by fuzzy aspects.

Parameters of Trust Evaluation. Specifically, trust should be quantifiable, measured for a particular context, and necessitated to be updated. In order to draw a comparative outline of the trust frameworks, Alhanahnah et al. [16] identified five criteria, based on the available literature.

- Standardization effort- whether or not trust frameworks contemplate cloud standards (such as 'Cloud Security Alliance's Cloud Controls Matrix' [41] and 'European Commission initiatives' [42]) in their design.
- Multifaceted criteria - measuring trustworthiness from diverse features, instead of relying only on a single percept for evaluation.
- Consumer perception of trust- determines whether the consumers are allowed to convey their orientation towards specific trust factors through the trust framework.
- Extensibility – ensures that the trust frameworks must be having the capacity to accommodate changes owing to the dynamic and constantly evolving nature of the cloud.
- Context awareness- whether the trust framework can comply to diverse application contexts with varying consumer's trust requirements.

4.3. Trust Value Computation and Evaluation Strategies. A simple method to compute reputation scores in a rating based system, as was used in eBay's reputation forum [43] [44] [45]. However, due to simplicity of this method the results obtained are not much effective. Most commercial websites like Epinions [46], eLance and Amazon use sophisticated approaches that calculate the weighted average over all ratings depending upon the user's credibility. Pan et al. [47] utilize Jaccard's Similarity Coefficient and Pearson Correlation Coefficient to seect trustorthy cloud services. Guo et al. [36] discuss several trust computation techniques that aggregate the trust values obtained from various sources. The authors reported weighted sum to be the most commonly used method in the literature. In reputations based systems [14], the most reputed users have the heaviest impact on the final trust value evaluated from the weighted sum method [48]. Other approaches described by Guo et al. [36] include Bayesian Systems [49] [50] [51], Regression Analysis [52], Belief Models [53], Fuzzy Models [54] [55] [56] [57] [40] [58] [29], and Flow Models [59] [60] [61]. Bayesian systems and Fuzzy models are the two most widespread trust evaluation strategies. Subjective logic, a special type of belief theory and regression analysis, has been mentioned in various studies [36]. Adopting logistic Regression analysis for trust estimation is a recent approach. It being more computation heavier than subjective logic provides better results. Subjective logic is based on the foundation of belief model. Figure 4.1 depicts the relation between the three variables defined by Josang [53], where the trust degree for an object i corresponds to a point in the triangle defined by the tuple

$$w_i = (b_i, d_i, u_i)$$

where *'belief' b* of an observer represents the belief in the object to be in trust state, *'disbelief' d* depicts the likelihood for the object not being in trust state and *'uncertainty' u* satiates the gap between *belief* and *disbelief*, as defined by Josang [53], such that

$$b + d + u = 1$$

Josang and Knapskog [62] gave a metric to evaluate the values of *b,d,u,* as depicted in Equations 4.1 to 4.3, where p and n represent the count of positive and negative experiences respectively.

$$b = \frac{p}{p+n+1} \tag{4.1}$$

$$d = \frac{n}{p+n+1} \tag{4.2}$$

$$u = \frac{1}{p+n+1} \tag{4.3}$$

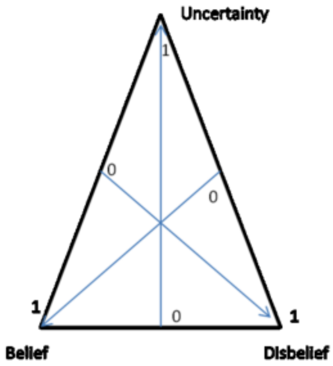

Fig. 4.1: Relationship between trust variables

Mei et al. [63] presented a 'Trusted Bytecode Virtual Machine Module' (TBVMM) as a technique of dynamic remote verification in cloud computing. To solve the dynamics of trust relationships in a distributed environment they used Bayesian model and Kalman filter. Wang et al. [64] investigated the existing dynamic trust level scheduling (DLS) and presented a cognitive model inspired novel Bayesian method for cloud computing. The proposed framework offered low dependability, integrity and safety confidentiality. Barsoum and Hasan [65] have proposed a cloud-based storage strategy that permits indirect reciprocal trust among the CSP and the data owner. Table 4.1 presents a detailed description of various trust computation techniques.

5. Cloud Trust Models. Trust is widely discussed in the generic computing literature, nonetheless, trust still is an emerging concept in the cloud computing environment. The challenges and issues in building trusted Cloud ecosystems has been talked about from various perspectives and there is a lack of any standard model for evaluating cloud trust. Trust is a standout amongst the most significant pointers for benefit determination and suggestion in Cloud adoption. There are three most common categories of trust models 1) Service Level Agreement (SLA) based trust model, 2) Recommendation based trust model and 3) Reputation based trust model. In this section we highlight some of the Cloud Trust models existing in the literature:

I. *TrustCloud Model*

Ryan K Lo and colleagues at Cloud & Security Lab of Hewlett-Packard Laboratories (HP), Singapore and Bristol presented the TrustCloud framework [10] that handles accountability issues in cloud computing through some technological and policy-based mechanisms. In their paper, they highlight that there is a pressing need for research in the domain of accountability in cloud systems. They classified trust components into two categories: Preventive Controls and Detective Controls. Preventive approach mitigates the happening of an event at all (like firewalls). Detective approach identifies the risks that are against the security and privacy policies of the system (like security audit trails, intrusion detection systems (IDS), and logs). Furthermore, there are also corrective controls, which happen to fix any unwanted events that occur.

For the cloud framework, the authors have focused on detective controls since they are non-invasive, and not only investigate the external risks alone, but the risks arising from inside the CSP as well. Although measures to directly stop the occurrence of irregularities are scarce, in cloud computing detective controls serve as some kind of a psychological barrier to policy breaches and even serve as a forensic record should there be any case of non-compliance. Detective controls act in a manner similar to speed cameras for traffic control. Generally, the combination of both the approaches is needed to complement each other for appropriate protection.

II. *EY Cloud Trust Model*

In the report "Building Trust in Cloud" Ernst & Young Global Ltd. (EY) [3] presented the 'EY Cloud Trust lifecycle Model' as a foundation that could be utilized by the organizations to build trusted cloud

Table 4.1: Description of various trust computation techniques

Technique	Description	Features	Source
Rating based system	Users provide a star rating as a feedback for products or services consumed.	Simplicity of method leading to ineffective results. Absence of feedback validity results in unfair ratings.	[43] [44] [45]
Bayesian Systems	Bayesian systems are built around prior beliefs for estimating the mean of a population.	This is useful for rankings so there is a need for a backdrop of a scale of all competing products' number of ratings. It is relevant when dataset is small.	[49] [50] [63] [56] [51]
Regression Analysis	A set of statistical processes for relationship estimation among variables. Mostly used for prediction and forecasting. Specifically it may be used to estimate continuous response variables.	Computation heavier than subjective logic, hence provides better results. Requires complex statistical analysis.	[42]
Belief Models /Subjective logic	It is a framework for dealing with uncertainty. It operates on opinions and beliefs about the world. A special type of belief theory and regression analysis is subjective logic.	It employs components from the 'Dempster-Shafer' belief theory. Uncertain probabilities are based on belief model.	[36] [53]
Fuzzy Models	Compute trust by fuzzy logic rules.	Need domain experts for doing parameter tuning and defining fuzzy rules	[54] [55] [56] [57] [40] [58]
Flow Models	Trust evaluation task is handled using network flow model.	Model is more compact and general and provides integral optimum solutions.	[59] [60] [61]
ACO (Ant Colony Optimization)	It introduces the pheromone concept and transition probability for dynamic trust representation.	Suitable for distributed feedback systems like cloud computing. Takes change of time and interaction frequency into account for direct trust evaluation.	[61] [62]

ecosystems. As shown in Fig. 5.1, the framework provides the following broad functionalities:

- Monitor and evaluate the risk profile of the organization and then developing strategies addressing the key areas of vulnerability.
- Improvise by performing remediation activities to enhance the developed strategies.
- Obtain certification and compliance from the third-party for assuring the security, trustworthiness, and auditability of the organization's cloud ecosystem.

The authors describe the six key dimensions that serve as a blueprint for the making of the trusted cloud systems [3]. The key dimensions also line up with the CSA's 'Cloud Control Matrix' and help in the understanding of the trust cloud characteristics. The six dimensions, as shown in Fig.5.1, are described below:

(a) Organizational: Organizations' internal users and CSP staff can introduce risk in the cloud ecosystem if the CSPs fail to manage organizational roles and human resources that deal with the challenges that emerge in a cloud ecosystem.

(b) Technology: With the right technical configurations of CSPs in place, like encryption, key management, access management, underlying infrastructure, vulnerability management, virtualization management, API security etc. can make a huge difference in building a trusted cloud ecosystem.

(c) Data: Maintaining the organization's data assets at different geographic locations puts challenges on

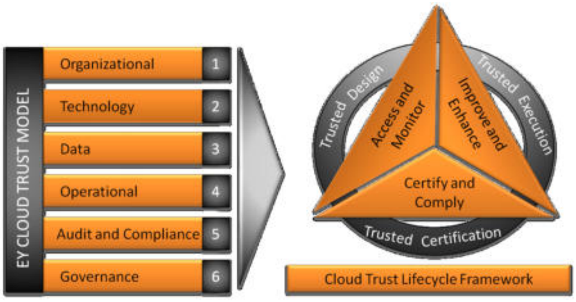

Fig. 5.1: EY Cloud Trust Model and Lifecycle Framework (after [3])

data protection. For adequate protection of data assets, organizations are required to know about their owned assets and the value that these assets hold.

(d) Operational: Moving IT operations to the CSP has an impact on those operations. There is a need of IT operation management system that controls the physical and environmental risks, provides backup and recovery plans, improved efficiency, and data security etc.

(e) Audit and compliance: CSPs need to have robust compliance procedures, audit plans and a third party assurance report at the minimum.

(f) Governance: A well-established governance model should be in place to design scalable programs, manage compliance and risks and monitor performance.

The authors conclude that by utilizing the ‘EY Cloud Trust model’ build upon six key cloud dimensions, trusted cloud ecosystems can be built. There has to be a balance between the risks and the value that the CSPs impart to the business. This trust model involves both preventive and detective measures for building trust.

III. *Hierarchical Trust Model*

In the paper [66], the authors presented a ‘Trust Model for Measuring the Security Strength of Cloud Computing Services’ that evaluates the cloud trust value based on the identified parameters relevant to security aspects. Both the static trust and the dynamic trust is evaluated to determine cloud service security. The evaluated trust value represents the overall strength of the cloud service security. Trust values identified include Identity Management System (IDM), Authentication, Authorization, Data Protection, Confidentiality, Communication, Isolation, Virtualization, and Compliance. The detailed description of the parameters is given in [66]. The trust model is indicated in the form of a hierarchical tree structure, where the root indicates the trust value and the child nodes represent the parameters with varying weights at different levels. The root node or the final trust value is the weighted sum of all the vector parameters. This model is based on the preventive approach of establishing trust.

Trust Factor Taxonomy:

In the article [16], the authors Alhanahnah et al. present a taxonomy of trust factors for trust evaluation that aims at helping cloud customers to choose trustworthy cloud service providers. This paper describes the notion of trust in cloud computing as discussed in Sec. 2. and also highlights some of the significant cloud trust concerns that demand rapid solutions for establishing and maintaining trust. Authors describe the two conceptual trust phases: 1) Establishing trust and 2) Maintaining trust. The establishing trust phase identifies the trust factors that evaluate the CSP trustworthiness. The trust factors are categorized as: 1) ‘SLA’ Trust Factors (TFs) and 2) ‘Non-SLA’ Trust Factors; as shown in Table.5.1. Briefly stated, SLA TFs are the trust properties extracted from the Service Level Agreements (SLA) [67] [68] [69] [70], whereas the non-SLA TFs are the trust properties derived from various other sources. The establishing trust phase generates a pretrust value which is then re-evaluated iteratively by the maintaining trust phase that follows; to keep the trust value updated. The maintaining trust phase monitors allegiance of service

Table 5.1: Taxonomy of Trust Factors (TFs)

SLA TFs	Non SLA TFs
Performance Security Data Management Personal Data Protection(PDP) Cost	CSCs Feedback Experts Opinion Financial Status

provider with the SLA which is significant for dynamic trust evaluation.

IV. *Trust Model in Cloud of Things Environment*
In the cloud of things environment, an autonomic system for trust management is difficult to be realized because there is a lack of control due to large scalability of deployment, node mobility, limited computing capacity [10] [71]. Therefore, in such scenarios, the trust manager must be flexible and adapt to the dynamic circumstances posed by the network. Namal et al. [4] present an autonomic trust framework for managing trust in Cloud-based IOT Applications called MAPE-K. MAPE-K is the feedback loop which is based upon ‘Monitor’, ‘Analyse’, ‘Plan’, ‘Execute’ and, ’Knowledge’ for evaluation of trust levels in IoT-cloud systems. The distributed trust agents in the framework filter the required trust information out of rest of the information to support autonomic trust decision-making. The monitor component filters and aggregates the information to detect any symptom which requires analysis. The analyze component carries out data analysis on the pre-detected symptoms from the monitor component. If any modifications are needed to achieve some targeted objectives, a change requisition is sent to the plan component that plans the actions and formulates the procedures needed to achieve the desired objectives. Execution phase actually brings the necessary modifications in the system behavior using effectors, as recommended by the plan component. The knowledge (K) includes the data associated with all the MAPE components that is centrally shared among all the trust agents in the network employing cloud infrastructure to serve decision-making. The knowledge data includes trust values, symptoms, context information, metrics, topology information, historical logs, and policies etc.

V. *Other Models*
M.Alhamad et.al. [40], presented a fuzzy logic based scheme for evaluating trust in cloud applications enabling cloud users to evaluate the trustworthiness of CSPs when migrating their operations to cloud data centers. The evaluation method uses four input factors: security, scalability, usability, and availability. The input parameters are fed to the fuzzy inference system employing Sugeno fuzzy technique with gbell membership function. Neural networks are employed to lessen the count of fuzzy rules in the training process to select only the most significant IF-THEN rules. M. Balasubramanian et al. [54] presented a framework to evaluate the trustworthiness of cloud servers and trust authority based on fuzzy approach by evaluating conformity on QoS parameters. Emeakaroha et.al [72] presented a trust-label system developed to communicate trust values in order to boost consumer confidences in Cloud services. Q. Chenhao and Rajkumar Buyya [56] presented a ‘Hierarchical Fuzzy Inference based System’ for evaluating Cloud Trust for Service Selection that evaluates trust of IaaS cloud based upon the user requirements. The model offers linguistic descriptors for both naïve and expert consumers to submit their vague demands or uncertain resources. The authors generated benchmark results for the 11 dynamic attributes that include CPU Speed, Memory Read, Memory Write, Disk Read, Disk Write, Network In, Network Out, Availability, Failure Rate, VM Startup, and VM Shutdown.
Sherchan et al. [73], proposed a Hidden Markov Model (HMM) based model for predicting trust in service web. It utilizes a time sensitive dynamic model for training pool. Xiaonian Wua et al. [74] proposed a ‘D-S evidence theory’ based model with sliding windows for evaluating trust in cloud computing. They calculated the direct trust of entities as the first-hand evidence by employing DS combination rules. Trust assesment relies upon the interaction evidence between CSP and CU. Since, the significance and legitimacy of evidence eventually would decay with time, sliding window is employed to delineate the timeliness of

Table 6.1: Comparison of different trust models

Source	Method / Model	Parameters	Observations
R.Shaikh [66]	Hierarchical tree structure based trust model for evaluating the overall strength of security in Cloud Computing Services .	Identified trust values include IDM, Authentication, Authorization, Data Protection, Confidentiality, Communication, Isolation, Virtualization, and Compliance.	Calculates both static and dynamic trust values. Uses weighted sum aggregation technique.
Ryan K. Lo [10]	TrustCloud framework	Preventive and Detective Controls	The model handles accountability issues of cloud systems through policy-based mechanisms.
Namal et al. [4]	An autonomic trust framework for managing trust in Cloud-based IOT Applications called MAPE-K.	The feedback loop which is based on ‘Monitor’, ‘Analyse’, ‘Plan’, ‘Execute’ and, ’Knowledge’ to evaluate trust.	MAPE-K evaluates the trust levels in an IoT cloud ecosystem.
M. Balasubramanian et al. [54]	Fuzzy logic approach	Evaluation based on conformity to QoS parameters.	The cloud server and trust authority parameter values are collected from cloud benchmark service.
M.Alhamad et al. [40]	Fuzzy logic based trust evaluation scheme for cloud applications.	Evaluation parameters include: security, scalability, usability, and availability.	Neural networks are employed to reduce the number of fuzzy rules in the training process to select only the most significant IF-THEN rule.
Xiaonian Wua et al. [74]	D-S evidence theory and sliding windows based trust evaluation model.	Direct trust of entities calculated by DS combination rules. Evaluation is based on interaction evidence between CSP and CU.	Describes the timeliness of the evidence information and takes care of the dynamisim of interaction evidences.
Lin et al. [75], Bedi et al. [76] [77]	Ant colony optimization (ACO) technique	Use ‘pheromone’ to model transition possibility for representing behavior trust.	ACO based trust-recommender system takes change of time and interaction frequency into account for trust evaluation.

the evidence information and it also takes care of the dynamism of interaction evidences.

Lin et al. [75] proposed a cloud trust model that is based on behavior of entities in cloud environment. Since the trust associations among corresponding entities are difficult, dynamic, and mostly unknown; the authors presented an Ant colony optimization (ACO) technique which is based on the conception of ‘pheromone’ to model transition possibility for representing behavior trust. Bedi et al. [76] also proposed an ant colony based trust-recommender system.

6. Discussion and Open Issues. The central aspect of trust management in Cloud environment is to determine how to assess the corresponding trust values of an entity. Various trust components have been specified however, not all possible trust properties should be gathered and stored. Determining the most relevant trust components is significant to obtain the accurate trust predictions. Machine learning approaches could be used to determine the set of properties that provide the most accurate trust predictions. Table 6.1 draws a comparison of different trust models and highlights the trust parameters identified in various studies.

As we explore some open issues in cloud trust management the central problem that has been identified is how to collect and verify trust information. The authenticity of trust information is a major concern and requires further research. Furthermore, different deployment models like public and private cloud models might need different trust management schemes. Trust information can be used as a metric to analyze system state in a particular duration or service context. Trust information can perform other tasks in addition to facilitating service selection. Lowered trust value could be an indication that the CSP is misbehaving due to resource-depletion at peak hours of the service usage. In the following subsection we provide a summary of the identified open issues that are potentially significant roadblocks that hinder cloud trust.

6.1. Open Issues. There is an absence of a reliable trust and reputation model that is standard and specific to the cloud architecture that assists the customers in choosing the trustworthy service providers [40]. Despite the fact that the development of trust and reputation management systems has been widespread and popularly implemented for several online services, there does not exist any such model implemented for cloud computing. Here we highlight some of the major open issues in Cloud Trust Management:

- As described in section 2, while we defined semantics of trust, trust is context-dependent. Hence, it may present some inaccurate information in some intense contextual conditions. There has to be a standard definition for describing trust in cloud scenario.
- The literature has addressed a broad range of SLA-based variables. However, there is no agreement defining and choosing a correct set of trust variables, as most methods are not in line with standardization bodies' norms and guidelines.
- Trust management system needs a uniform mechanism for accumulating multiple trust attributes irrespective of the evaluation procedures employed to evaluate the subjective trust parameters.
- In the complex IoT-based cloud environments, SLAs are inadequate. Often times, the ambiguous clauses and vague technical specifications of SLAs can prevent service consumers from identifying trustworthiness of cloud services.
- In the cloud of things environment, having an autonomic trust management system is difficult to be realized due to node mobility, limited computing capacity, and lack of control [10] [71].

6.2. Datasets for Trust Evaluation. We have identified some datasets that could be employed to evaluate trust related computations applicable to cloud computing setting:

- *Cloud Armor dataset* [78]: It is the real time data containing the trust information obtained from various cloud service providers derived from consumer feedbacks [32].
  [Cloud Armor [79] is a research project aiming to develop a scalable Trust Management System for cloud services, at the University of Adelaide, Australia.]
- *Trust Feedback Dataset* [78]: This dataset is a collection of consumers' feedback of cloud services; based on QoS attributes. The dataset is a collection of 10,000+ feedbacks received from nearly 7,000 service consumers for 113 real-world cloud services.
- *Epinion Dataset* [80]: Epinion is a review website where people post product reviews. A trust network of users is created by adding other users to their ”Web of Trust” whose reviews and ratings they find consistently valuable” and conversely adding users to their ”Block list” whose reviews are found consistently not valuable or inaccurate. The Epinion dataset collected from the website Epinions.com contains about 664,824 reviews and 487,181 issued trust statements from 49,290 users for 139,738 different items.
- *Extended Epinion Dataset* [80]: This dataset also contains distrust lists for items. The dataset consists of 841,372 reviews (717,667 trusts and 123,705 distrusts) received from about 132,000 users.

The datasets described above are merely based on user feedbacks and rely solely on user experiences and preferences. These consumer feedbacks act as Trust Indicators that lead to Trust Formation and assist cloud users to choose a trustworthy cloud service among a pool of available service providers; and also gives cloud users an idea of the services they can trust. However, there is a lack of available data for evaluating cloud services based on their functionalities rather than the user experiences or ratings. Evaluating services for their functionalities provides more transparent, accurate and direct trust metrics for system evaluation. We derived a service evaluation dataset from a standard BOSSBase dataset for trust verification of cloud services, the details of which are introduced in the next section.

7. Proposed Rich Model based Machine Learning solution for Trust Verification. As the advancements in the field of Machine Learning continue to expand, several approaches to solving cloud computing challenges have utilized the opportunities of problem solving using Machine Leaning algorithms [81] [82]. Gulen et al. [83], [84] utilized machine learning algorithms to solve the issue of anomaly detection in Clouds. Khilar et al. [85] put forth an access control model for cloud computing based on trust by employing Machine Learning. Wang et al. [86] designed a machine learning techniques based privacy-preserving framework using feature extraction.

In this study we identified one major direction that would serve as a roadmap for creating trusted cloud environment. As we have seen that cloud-based services have increasingly gained much popularity, yet there is a lack in the tools that allow cloud consumers to verify that these services perform as expected [87] [88] [89]. Dykstra et al. [90] for the first time provided an evaluation model for some cloud-forensic acquisition tools that aids in providing confidences in the acquired evidences. In addition to promising security guarantees these tools should verify functional correctness and performance along with the service availability and reliability. Users would be immensely benefited from knowing as to what extent the CSP delivered the services as promised. More precisely, the agreement between the cloud consumer and the service provider should be verified and not merely be reliant on cloud provider's report. It is quiet important to capture the misbehavior of CSP in terms of functional correctness of the service promised so as to detect failures. As cloud software is delivered by a third party as a Software-as-a-Service, the actual code of implementation is expected to be unavailable to the service users. Thus the utilization of white-box technique strategies (like symbolic execution [91] [92] [93]) cannot be utilized.

In our work we mainly focus on two key areas: A) Trusted Software: To verify whether the right software is running for the correct services. B) Functional correctness: To verify whether the running service is working as it is expected to perform. Here we consider cloud customers falling into two categories, one is the service providers that deploy their services or softwares in the cloud (PaaS or IaaS), and the other is cloud users that utilize a software or cloud storage service.

7.1. Approach for verifying the trusted software. If the software deployed by the cloud service provider is tampered with or replaced by some other low cost version, the requested service being invoked by the user would deviate from the expected behavior and result in trust violation. Hence, for the service providers in order to guarantee that trusted software is running, verifying the trusted software running on cloud nodes on the basis of output generated from the service utilization be facilitated at the consumer end. We propose an approach to verify functional properties of a cloud services based on the validation of results or the output generated after utilizing a particular cloud service; without involving a third party for certification or without the need of accessing the implementation code of the service utilized. As image processing jobs are computation and storage costly services, due to their large scale processing needs, many such requests are processed over the cloud to reduce computational and implementation overheads at the client site.

As a use-case, we consider a scenario where one such service is being invoked by a cloud user. The user requests the service over a set of input images submitted to a cloudlet. The service provider performs the computation over a given set of input data as requested by the user. At this point the service provider may either satisfy the user request by faithfully processing the data utilizing the legitimate software or maliciously employ a low-cost, or some tampered/ unlicensed version of the software. The users would find it difficult to capture the violation in the services delivered unless they have the verification mechanisms at their disposal to verify the legitimacy of the output produced. We have proposed a Machine Learning backed software verification approach based on Spatial Rich Models (SRM) [94] inspired by its wide applicability in steganographic detectors in the domain of digital forensics [95] [96] [97] [98].

Our approach relies on the hypothesis that any data processing service would leave some digital footprints that would enable verification of the software utilized for processing user requests. The goal is to extract distinct features from the processed output returned to the service consumer in order to capture the slight differences in the output generated from the distinct softwares utilized by the service provider. We extract spatial rich features for the output data and investigate the residual noise distributions for capturing any violations in the usage of software-as-a-service. We evaluate these features by the Machine learning model called Ensemble Classifier [99] which constitutes an array of base learners that predicts the output class based on majority voting. The effectiveness of the classifier is evaluated using ROC curve. The detailed algorithms for generating features for evaluation and model training are explained in the Algorithm 1 and 2 respectively.

```
Algorithm 1: Generate Features for verification $(D_n, F)$
    Input: $\boldsymbol{D}_n$ Set of Input data
    Output: $\boldsymbol{F}$ Feature Set
        1. User submits $\boldsymbol{n}$ service requests to cloud server:
            I. Cloud server receives $\boldsymbol{D}_I = (\boldsymbol{D}_{i1}, \boldsymbol{D}_{i2}, \boldsymbol{D}_{i3} \ldots \boldsymbol{D}_{in})$ input n files to be processed.
        2. Cloud service provider invokes the requested service to process input queries.
            I. Apply the required data processing operations on the input data received $\boldsymbol{D}_I$.
               (At this point the service provider may ideally utilize the genuine software to process the requests
               and faithfully return the output data to the user or may maliciously utilize a low cost version of
               the software for the task.)
        3. Return the output data to the customer after applying the required data processing.
            I. Customer receives the resulting processed data $\boldsymbol{D}_R = (\boldsymbol{D}_{r1}, \boldsymbol{D}_{r2}, \boldsymbol{D}_{r3} \ldots \boldsymbol{D}_{rn})$, from the Cloud
               Server.
            II. This $\boldsymbol{D}_R = (\boldsymbol{D}_{r1}, \boldsymbol{D}_{r2}, \boldsymbol{D}_{r3} \ldots \boldsymbol{D}_{rn})$, will be used to verify the cloud service utillized.
        4. Extract spatial rich features in the output data $\mathbf{D}_R$ by employing Spatial Rich Models (SRM).
            I. Extract the discrete rich models that are constructed from the neighbouring noise residual samples
               in the spatial domain.
            II. Extract all the 106 rich spatial submodels from the output data returned by the service provider;
                for each data item in $\boldsymbol{D}_R$.
            III. The extracted submodels ($\boldsymbol{F}$) have the dimensionality of 34671, for each $\boldsymbol{R}$ in $\boldsymbol{D}_R$.
            IV. Spatial rich features $\boldsymbol{F} = (\boldsymbol{F}_1, \boldsymbol{F}_2, \boldsymbol{F}_3 \ldots \boldsymbol{F}_n)$, are analyzed to verify the legitimacy of the cloud
                service utilized.
```

```
Algorithm 2: Evaluating Features for Software Verification $(F, C_p)$
    Input: $\boldsymbol{F}$ Feature Set
    Output: $\boldsymbol{C}_p$ Predicted Class
        1. Train the Machine learning model for Feature Evaluation.
            I. Train the verification model on random samples ($\boldsymbol{F}$) of the features obtained from pre-processed
               data of licensed and unlicensed versions, using cross validation method.
            II. Decision is carried out by majority voting by employing Ensemble classifier of $\boldsymbol{L}$ binary base
                learners.
        2. Verify the cloud service utilized by the customer
            I. Customer verifies the data $\boldsymbol{D}_R$ received from the Cloud Server for its legitimacy using the trained
               machine learning classification model.
            II. The area-under-curve (AUC) in the ROC curve gives the classification prediction accuracy of the
                model.
```

7.2. Experimental Setup. For the usecase described in Section 7, we utilized SRM for feature extraction as it has been widely employed to extract spatial features from neighboring pixels to capture slight variations in the spatial domain. The process involves assembling many submodels derived from the noise distributions of neighboring samples of image residuals. The assembling of submodels is made a part of the machine learning training process that is driven by corresponding Matlab and Octave processed samples. By capturing the variations in the spatial domain, after the data undergoes processing using different softwares, we were able to utilize the power of rich models for verifying trust in cloud services. Another important aspect of employing SRM as a feature extractor is its ability to be used as a general-purpose model for digital forensics as it is independent of the data content [94].

As a feature classifier, and for assembling individual models, we utilized Ensemble classifier. As SRM features have a high dimentionality and we are processing large training datasets, the random forest of Ensemble classifiers is more suitable due to its low computational complexity and its efficiency for high-dimensional features involving big training datasets. Using Ensemble as a classifier also solves the problem of over-fitting when dealing with large training samples; which could have been a case if we used SVM classifier instead. The Ensemble classifier is based on an array of individual base learners that arrive at a consensus by majority voting. Every individual learner is trained on a random feature-set and acts as an independent classifier. By evaluating how various spatial submodels involve in trust verification of cloud services, trust violation is detected.

Our proposed model works in two steps: A) Extracting features for model training and service verification using SRM. B) Evaluating rich features for service verification using Ensemble classifier. First off we generate service utilization datasets that act as samples to pre-train our proposed model. The datasets were created by performing image processing operations on a standard BOSSBase dataset. To verify the effectiveness of our model we invoked image processing services using a legitimate software(Matlab) and also using is low-cost version (Octave) for capturing variations in spatial domain. After deriving the required datasets for Matlab and Octave, we extracted rich features for the derived datasets using SRM. A total of 106 sub-models are extracted, each having an average dimentionality of 327, thereby making a total dimentionality of 34671 for a single feature set against each data file. Then, we trained random forest of Ensemble classifiers using ten-fold cross validation on the extracted rich feature-sets.

Once our model is trained on sufficient feature-sets, it is utilized for trust evaluation of services utilized. When a cloud user submits a request along with the input data to a cloud server invoking a service, the service provider responds with the requested output. Whether the service provider has utilized the legitimate software for processing the user request, is verified by evaluating the rich features of the output produced.

Due to the lack of availability of Trust-datasets that evaluate direct trust values in cloud services, we derived a new trust dataset for direct trust evaluation. The datasets described in section 6.2 include user feedbacks and reviews, hence these can be utilized to evaluate indirect trust that does not add to increased transparency in cloud service monitoring.

7.3. Results and Discussion. As a usecase we consider an image processing task being invoked by the customer wherein the user requests for Edge Detection algorithm operated over a set of input images using Matlab software. We extract two streams of SRM features, one corresponding to a set of images processed by Matlab and another using an unlicensed low cost version Octave. We have explained the details of the experiments in [100] and [101]. We utilized the standard BOSSBase dataset having 10,000 images, for our experimentation. The prediction accuracy of the trained Ensemble classifier is calculated by out-of-bag (OOB) error. Figure 7.1. a) shows the histogram of base learners and Fig 7.1. b) shows the variation in OOB estimate with variation in the count of base learners. The count of base learners, at which the OOB saturates, is fixed. Fig 7.2. shows the area-under-curve plot of ROC curve that gives the true positive rate (recall) vs. false positive rate (fall-out). The results suggest that spatial rich features give a prediction accuracy of 99% with an average testing error of 0.0040 (+/- 0.0009) calculated over 10 splits. This high prediction accuracy is accredited to the large number of diverse submodels involved in extracting rich features using SRM, along with all the varying combinations of submodels and their quantization levels. These individual models capture large number of relationships among neighboring pixels in the data and thus contribute to high prediction accuracy.

Although, Rich Models have not been utilized for evaluating cloud service trust verification, but we have compared its prediction accuracy with related domains that utilize SRM. Jian et.al [102] utilized local residual descriptors for predicting recapture effect in images and achieved an accuracy of 96.72% with three submodels and an accuracy of 98.43% with ten submodels of SRM. Han et.al [95] utilized Rich models for detecting image manipulations. They adopted two-stream R-CNN network to predict if the image has been manipulated or not and achieved a prediction accuracy of 93.7% on NIST16 dataset. In another similar work Cozzolino et.al. [103] utilized SRM for image forgery detection. They extracted a single model SRM feature-set using CNN and employed SVM classifier for forgery detection. The prediction accuracy recorded was 84% to 99% for different image manipulations.

These results suggest that Rich Models are powerful feature extractors that have a great potential in the domain of image forensics and we have successfully utilized its potential for verifying cloud services for functional correctness. The proposed approach of feature generation from the output data for verifying the cloud trust

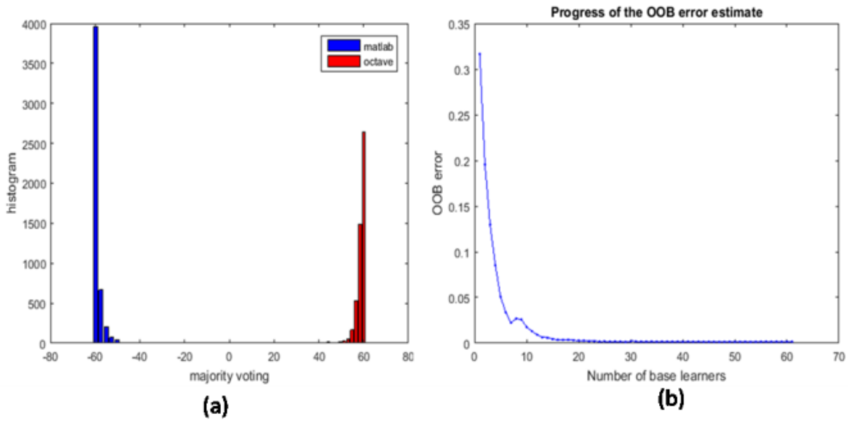

Fig. 7.1: a) Majority voting histogram b) Change of OOB error with change in base learner count.

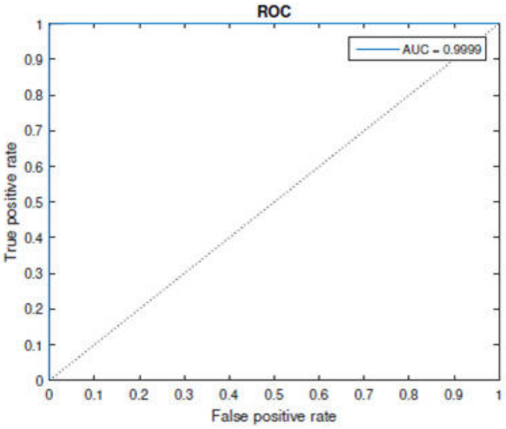

Fig. 7.2: Classification accuracy ROC estimate.

in delivering services as expected can be perceived as an initiative towards trust evaluation in cloud services employing Machine learning techniques.

8. Conclusion. Cloud computing being an opportune technology, but inadequate trust management is hindering its progress. Despite significant efforts to mitigate trust issues, the lack of control and fear of change inhibits individuals and organizations to adopt cloud computing. Even though the legal approaches have been laid down for cloud trust assurance, they continue to be insufficient on their own. Cloud service users still have to rely on cloud providers' promises to provide the desired services. Rather than obliging consumers to rely on providers' genuine behavior, cloud services should employ a standard trust management system so that the users could access and predict accurate trust information. In the literature, the detailed review discussing the issue of trust management is very rare. We presented a review of trust management in the cloud context and highlighted some open issues and possible research domains to uncover the trust issues in cloud computing scenario. A few open questions have been identified. The central issue is to determine the most relevant trust properties to predict accurate trust values and to identify the most effective method to aggregate multiple trust variables to obtain the final trust value. One open issue is to reach for a trade-off between a centralized and distributed trust propagation technique. A potential solution is to implement a hierarchical approach by combining both the approaches. We proposed a Rich model based Machine Learning backed solution to verify the software-as-a-service utilized by the cloud consumer. We performed the experimentation on the standard dataset to evaluate the effectiveness of our scheme to verify the utilization of spatial noise residuals as features for classifying trusted services. Nevertheless, there are several open questions pertinent to trust management in cloud computing that need to be explored further.

REFERENCES

[1] N. Santos, R. Rodrigues, K. P. Gummadi, S. Saroiu, Policy-Sealed Data: A New Abstraction for Building Trusted Cloud Services, Tech. rep.

[2] P. Mell, T. Grance, The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology, Nist Spec. Publ. 145 (2011) 7. arXiv:2305-0543, doi:10.1136/emj.2010.096966.

[3] EY, Building trust in the cloud Creating confidence in your cloud ecosystem, Insights governance, risk compliance (June) (2014).

[4] S. Namal, H. Gamaarachchi, G. M. Lee, T. W. Um, Autonomic trust management in cloud-based and highly dynamic IoT applications, in: Proc. 2015 ITU Kaleidosc. Trust Inf. Soc. K-2015 - Acad. Conf., IEEE, 2016, pp. 1–8. doi:10.1109/Kaleidoscope.2015.7383635.

[5] K. M. Khan, Q. Malluhi, Establishing Trust in Cloud Computing, IT Prof. 12 (5) (2010) 20–27. doi:10.1109/MITP.2010.128.

[6] M. S. Khan, M. R. Warsi, S. Islam, Trust Management Issues in Cloud Computing Ecosystems, in: Elsevier SSRN Ser., Elsevier, 2019, pp. 2233–2238.

[7] T. Lynn, P. Healy, R. McClatchey, J. Morrison, C. Pahl, B. Lee, The Case for Cloud Service Trustmarks and Assurance-as-a-Service, arXiv Prepr. (feb 2014). arXiv:1402.5770.

[8] M. Tang, X. Dai, J. Liu, J. Chen, Towards a trust evaluation middleware for cloud service selection, Futur. Gener. Comput. Syst. 74 (2017) 302–312. doi:10.1016/j.future.2016.01.009.

[9] N. Gonzalez, C. Miers, F. Red, M. Simpl, Open Access A quantitative analysis of current security concerns and solutions for cloud computing (2012) 1–18.

[10] R. K. Ko, P. Jagadpramana, M. Mowbray, S. Pearson, M. Kirchberg, Q. Liang, B. S. Lee, TrustCloud: A framework for accountability and trust in cloud computing, in: Proc. - 2011 IEEE World Congr. Serv. Serv. 2011, 2011, pp. 584–588. doi:10.1109/SERVICES.2011.91.

[11] J. B. Michael, L. T. Gaines, Trust Management in Distributed Databases, in: Computer (Long. Beach. Calif)., Vol. 40, IEEE Computer Society Press, 2005, pp. 329–337. doi:10.1007/0-306-47008-x_29.

[12] T. S. Dybedokken, Trust Management in Fog Computing, Futur. Gener. Comput. Syst. 5 (June) (2017) 15619–15629. doi:10.1109/ACCESS.2017.2733225.

[13] D. M. Rousseau, S. B. Sitkin, R. S. Burt, C. Camerer, Not so different after all: A cross-discipline view of trust, Acad. Manag. Rev. 23 (3) (1998) 393–404. doi:10.5465/AMR.1998.926617.

[14] A. Jøsang, R. Ismail, The beta reputation system, 15th Bled Electron. Commer. Conf. (2002) 2502–2511doi:10.1.1.60.5461.

[15] A. Jøsang, C. Keser, T. Dimitrakos, Can we manage trust?, in: Lect. Notes Comput. Sci., Vol. 3477, Springer, Berlin, Heidelberg, 2005, pp. 93–107. doi:10.1007/11429760_7.

[16] M. Alhanahnah, P. Bertok, Z. Tari, Trusting cloud service providers: Trust phases and a taxonomy of trust factors, IEEE Cloud Comput. 4 (1) (2017) 44–54. doi:10.1109/MCC.2017.20.

[17] P. Massa, K. Souren, Trustlet, open research on trust metrics, in: CEUR Workshop Proc., Vol. 333, 2008, pp. 31–44.

[18] Fujitsu, Personal data in the cloud: (2010).
URL `https://www.fujitsu.com/ie/imagesgig5/fujitsu`$_{p}ersonal - data - in - the - cloud.pdf$

[19] Service Level Agreement - Amazon Simple Storage Service (S3).pdf.
URL `https://aws.amazon.com/s3/sla/`

[20] Amazon EC2 Service Level Agreement – Amazon Web Services.
URL `https://aws.amazon.com/ec2/sla/historical/`

[21] Service Level Agreements - Home | Microsoft Azure.
URL `https://azure.microsoft.com/en-in/support/legal/sla/`

[22] Cloud Security Alliance, CSA - Top Threaths to Cloud Computing v1.0, Security (March) (2010) 1–14.
URL `https://ioactive.com/wp-content/uploads/2018/05/csathreats.v1.0-1.pdf`

[23] J. Schiffman, T. Moyer, H. Vijayakumar, T. Jaeger, P. McDaniel, Seeding clouds with trust anchors, in: Proc. ACM Conf. Comput. Commun. Secur., ACM Press, New York, New York, USA, 2010, pp. 43–48. doi:10.1145/1866835.1866843.

[24] M. Abomhara, G. M. Kien, Cyber Security and the Internet of Things: Vulnerabilities, Threats, Intruders and Attacks, J. Cyber Secur. Mobil. 4 (1) (2015) 65–88. doi:10.13052/jcsm2245-1439.414.

[25] J. Harauz, B. P. Lori M. Kaufman, Data Security in the World of Cloud Computing, IEEE Secur. Priv. (2009) 61–64.

[26] T. H. Noor, Q. Z. Sheng, Z. Maamar, S. Zeadally, Managing Trust in the Cloud: State of the Art and Research Challenges, IEEE Comput. 49 (2) (2016) 34–45. doi:10.1109/MC.2016.57.

[27] M. Blaze, J. Feigenbaum, J. Lacy, Decentralized trust management, in: Proc. 1996 IEEE Symp. Secur. Priv., 1996, pp. 164–173. doi:10.1109/SECPRI.1996.502679.

[28] C. V. L. Mendoza, J. H. Kleinschmidt, Mitigating on-off attacks in the internet of things using a distributed trust management scheme, Int. J. Distrib. Sens. Networks 2015 (2015). doi:10.1155/2015/859731.

[29] L. Gu, J. Wang, B. Sun, Trust management mechanism for Internet of Things, China Commun. 11 (2) (2014) 148–156. doi:10.1109/CC.2014.6821746.

[30] I.-R. Chen, F. Bao, J. Guo, Trust-Based Service Management for Social Internet of Things Systems, IEEE Trans. Dependable Secur. Comput. 13 (6) (2016) 684–696. doi:10.1109/TDSC.2015.2420552.

[31] M. Alruwaythi, K. Kambhampaty, K. E. Nygard, User Behavior and Trust Evaluation in Cloud Computing 58 (2019) 378–368. doi:10.29007/q5bd.

[32] Cloud Armor Project Website - About.
URL `https://cs.adelaide.edu.au/ cloudarmor/research.html`

[33] M. Chiregi, N. J. Navimipour, A comprehensive study of the trust evaluation mechanisms in the cloud computing, J. Serv. Sci. Res. 9 (1) (2017) 1–30. doi:10.1007/s12927-017-0001-7.

[34] R. Marty, Cloud application logging for forensics, 2011, p. 178. doi:10.1145/1982185.1982226.

[35] J. R. Jain, A. Asaduzzaman, A Novel Data Logging Framework to Enhance Security of Cloud Computing (2016).

[36] J. Guo, I. R. Chen, J. J. Tsai, A survey of trust computation models for service management in internet of things systems, Comput. Commun. 97 (2017) 1–14. doi:10.1016/j.comcom.2016.10.012.

[37] H. Kim, H. Lee, W. Kim, Y. Kim, A trust evaluation model for QoS guarantee in cloud systems, Int. J. Grid Distrib. ... 3 (1) (2010) 1–10.
URL `http://www.sersc.org/journals/IJGDC/vol3_no1/1.pdf`

[38] Y. Wang, J. Wen, X. Wang, B. Tao, W. Zhou, A cloud service trust evaluation model based on combining weights and gray correlation analysis, Secur. Commun. Networks 2019 (2019). doi:10.1155/2019/2437062.

[39] Y. Chen, R. H. Katz, What ' s New About Cloud Computing Security ? (2010).

[40] M. Alhamad, T. Dillon, E. Chang, A Trust-Evaluation Metric for Cloud applications, Int. J. Mach. Learn. Comput. 1 (4) (2013) 416–421. doi:10.7763/ijmlc.2011.v1.62.

[41] Cloud Controls Matrix | Cloud Security Alliance.
URL `https://cloudsecurityalliance.org/working-groups/cloud-controls-matrix/#_overview`

[42] E. Commission, The European Cloud Initiative | Digital Single Market (2018).
URL `https://ec.europa.eu/digital-single-market/en/ european-cloud-initiative`

[43] Ebay, Feedback.
URL `https://pages.ebay.com/services/forum/feedback.html`

[44] B. Rietjens, Trust and reputation on eBay: Towards a legal framework for feedback intermediaries, Inf. Commun. Technol. Law 15 (1) (2006) 55–78. doi:10.1080/13600830600557935.
URL `http://www.tandfonline.com/action/journalInformation?journalCode=cict20`

[45] P. Y. Sun, Building Trust in Online Rating Systems through Signal Modeling Online Feedback - based Rating Systems, Strategy.

[46] Shopping, Shopping Online at Shopping.com | Price Comparison Site.
URL `http://epinion.com/?sb=1`

[47] Y. Pan, S. Ding, W. Fan, J. Li, S. Yang, Trust-enhanced cloud service selection model based on QoS analysis, PLoS One 10 (11) (2015) 1–14. doi:10.1371/journal.pone.0143448.

[48] V. Shmatikov, C. Talcott, Reputation-based trust management, J. Comput. Secur. 13 (1) (2005) 167–190. doi:10.3233/JCS-2005-13107.

[49] D. Marudhadevi, V. N. Dhatchayani, V. S. S. Sriram, A Trust Evaluation Model for Cloud Computing Using Service Level Agreement, Comput. J. 58 (10) (2014) 2225–2232. doi:10.1093/comjnl/bxu129.

[50] H. T. Nguyen, W. Zhao, J. Yang, A trust and reputation model based on Bayesian network for web services, in: ICWS 2010 - 2010 IEEE 8th Int. Conf. Web Serv., IEEE, 2010, pp. 251–258. doi:10.1109/ICWS.2010.36.

[51] L. Wang, X. Li, X. Yan, S. Qing, Y. Chen, Service Dynamic Trust Evaluation Model based on Bayesian Network in Distributed Computing Environment 9 (5) (2015) 31–42.

[52] Y. Wang, Y.-c. Lu, I.-r. Chen, J.-h. Cho, A. Swami, C.-t. Lu, LogitTrust : A Logit Regression-based Trust Model for Mobile Ad Hoc Networks State of the Art, Proc. 6th ASE Int. Conf. Privacy, Secur. Risk Trust (PASSAT '14) (2014).

[53] A. Jøsang, A Logic for Uncertain Probabilities, Int. J. Uncertainty, Fuzziness Knowledge-Based Syst. 09 (03) (2001) 279–311.

[54] M. Balasubramanian, H. Kim, Trust evaluation scheme for cloud data security using fuzzy based approach, Int. J. Appl. Eng. Res. 12 (13) (2017) 3908–3913.

[55] X. Anita, M. A. Bhagyaveni, J. M. L. Manickam, Fuzzy-Based trust prediction model for routing in WSNs, Sci. World J. 2014 (iii) (2014). doi:10.1155/2014/480202.

[56] C. Qu, R. Buyya, A cloud trust evaluation system using hierarchical fuzzy inference system for service selection, in: Proc. - Int. Conf. Adv. Inf. Netw. Appl. AINA, no. May, 2014, pp. 850–857. doi:10.1109/AINA.2014.104.

[57] P. N. Mahalle, P. A. Thakre, N. R. Prasad, R. Prasad, A fuzzy approach to trust based access control in internet of things, in: 2013 3rd Int. Conf. Wirel. Commun. Veh. Technol. Inf. Theory Aerosp. Electron. Syst. VITAE 2013 - Co-located with Glob. Wirel. Summit 2013, IEEE, 2013, pp. 1–5. doi:10.1109/VITAE.2013.6617083.

[58] J. Du, X. Li, Adaptive and attribute-based trust model for service-level agreement guarantee in cloud computing, IET Inf. Secur. 7 (1) (2013) 39–50. doi:10.1049/iet-ifs.2012.0232.

[59] C. N. Ziegler, G. Lausen, Spreading activation models for trust propagation, in: Proc. - 2004 IEEE Int. Conf. e-Technology, e-Commerce e-Service, EEE 2004, IEEE, 2004, pp. 83–97. doi:10.1109/EEE.2004.1287293.

[60] S. Brin, L. Page, The PageRank Citation Ranking: Bringing Order to the Web, BMC Syst. Biol. 4 Suppl 2 (1999-66) (2010) S13. arXiv:1111.4503,

[61] W. Jiang, J. Wu, F. Li, G. Wang, H. Zheng, Trust evaluation in online social networks using generalized network flow, IEEE Trans. Comput. 65 (3) (2016) 952–963. doi:10.1109/TC.2015.2435785.

[62] A. Jøsang, A. Jøsang, S. J. Knapskog, A Metric for Trusted Systems, Proc. 21ST Natl. Secur. Conf. NSA (1998).
URL `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.38.8631`

[63] S. Mei, Z. Wang, Y. Cheng, J. Ren, J. Wu, J. Zhou, Trusted Bytecode Virtual Machine Module: A Novel Method for Dynamic Remote Attestation in Cloud Computing, Int. J. Comput. Intell. Syst. 5 (5) (2012) 924–932. doi:10.1080/18756891.2012.733231.

[64] W. Wang, G. Zeng, D. Tang, J. Yao, Cloud-DLS: Dynamic trusted scheduling for Cloud computing, Expert Syst. Appl. 39 (3) (2012) 2321–2329. doi:10.1016/j.eswa.2011.08.048.

[65] A. Barsoum, A. Hasan, Enabling Dynamic Data and Indirect Mutual Trust for Cloud Computing Storage Systems, IEEE Trans. Parallel Distrib. Syst. 24 (12) (2013) 2375–2385. doi:10.1109/TPDS.2012.337.

[66] R. Shaikh, M. Sasikumar, Trust model for measuring security strength of cloud computing service, in: Procedia Comput. Sci., Vol. 45, Elsevier Masson SAS, 2015, pp. 380–389. doi:10.1016/j.procs.2015.03.165.

[67] M. B. Chhetri, Q. B. Vo, R. Kowalczyk, Policy-based automation of SLA establishment for cloud computing services, in: Proc. - 12th IEEE/ACM Int. Symp. Clust. Cloud Grid Comput. CCGrid 2012, IEEE, 2012, pp. 164–171. doi:10.1109/CCGrid.2012.116.

[68] M. Macías, J. Guitart, Client classification policies for SLA enforcement in shared cloud datacenters, in: Proc. - 12th IEEE/ACM Int. Symp. Clust. Cloud Grid Comput. CCGrid 2012, 2012, pp. 156–163. doi:10.1109/CCGrid.2012.15.

[69] L. Malrait, S. Bouchenak, N. Marchand, Experience with CONSER: A system for server control through fluid modeling, IEEE Trans. Comput. 60 (7) (2011) 951–963. doi:10.1109/TC.2010.164.

[70] V. C. Emeakaroha, M. A. Netto, R. N. Calheiros, I. Brandic, R. Buyya, C. A. De Rose, Towards autonomic detection of SLA violations in Cloud infrastructures, Futur. Gener. Comput. Syst. 28 (7) (2012) 1017–1029. doi:10.1016/j.future.2011.08.018.

[71] S. Pearson, A. Benameur, Privacy, security and trust issues arising from cloud computing, in: Proc. - 2nd IEEE Int. Conf. Cloud Comput. Technol. Sci. CloudCom 2010, IEEE, Pearson2010, 2010, pp. 693–702. doi:10.1109/CloudCom.2010.66.

[72] V. C. Emeakaroha, K. Fatema, L. Van Der Werff, P. Healy, T. Lynn, J. P. Morrison, A Trust Label System for Communicating Trust in Cloud Services, IEEE Trans. Serv. Comput. 10 (5) (2017) 689–700. doi:10.1109/TSC.2016.2553036.

[73] W. Sherchan, S. Nepal, A. Bouguettaya, A trust prediction model for service web, Proc. 10th IEEE Int. Conf. Trust. Secur. Priv. Comput. Commun. Trust. 2011, 8th IEEE Int. Conf. Embed. Softw. Syst. ICESS 2011, 6th Int. Conf. FCST 2011 (2011) 258–265doi:10.1109/TrustCom.2011.35.

[74] X. Wu, R. Zhang, B. Zeng, S. Zhou, A trust evaluation model for cloud computing, in: Procedia Comput. Sci., Vol. 17, 2013, pp. 1170–1177. doi:10.1016/j.procs.2013.05.149.

[75] G. Lin, Y. Bie, M. Lei, K. Zheng, ACO-BTM: A Behavior Trust Model in Cloud Computing Environment, Int. J. Comput. Intell. Syst. 7 (4) (2014) 785–795. doi:10.1080/18756891.2013.864479.

[76] P. Bedi, R. Sharma, Trust based recommender system using ant colony for trust computation, Expert Syst. Appl. 39 (1) (2012) 1183–1190. doi:10.1016/j.eswa.2011.07.124.

[77] J. Bharath, V. S. S. Sriram, Genetically Modified Ant Colony Optimization based Trust Evaluation in Cloud Computing 9 (December) (2016). doi:10.17485/ijst/2016/v9i48/107967.

[78] Cloud Armor Project Website - Dataset.
URL `https://cs.adelaide.edu.au/ cloudarmor/ds.html`

[79] T. H. NOOR, S. QUAN Z., S. ZEADALLY, Y. U. JIAN, Trust Management of Services in Cloud Environments: Obstacles and Solutions., ACM Comput. Surv. 46 (1) (2013) 12 – 12:30. doi:10.1145/2522968.2522980.

[80] Epinions trust network dataset – {KONECT} (2017).
URL `http://www.trustlet.org/epinions.html http://konect.uni-koblenz.de/networks/epinions`

[81] J. Fiala, A Survey of Machine Learning Applications to Cloud Computing (2015) 1–26.
URL `http://www.cse.wustl.edu/ jain/cse570-15/ftp/cld_ml.pdf`

[82] D. Pop, Machine Learning and Cloud Computing: Survey of Distributed and SaaS Solutions (2016). arXiv:1603.08767.
URL `http://arxiv.org/abs/1603.08767`

[83] A. Gulenko, M. Wallschlager, F. Schmidt, O. Kao, F. Liu, Evaluating machine learning algorithms for anomaly detection in clouds, Proc. - 2016 IEEE Int. Conf. Big Data, Big Data 2016 (2016) 2716–2721doi:10.1109/BigData.2016.7840917.

[84] T. Salman, D. Bhamare, A. Erbad, R. Jain, M. Samaka, Machine Learning for Anomaly Detection and Categorization in Multi-Cloud Environments, Proc. - 4th IEEE Int. Conf. Cyber Secur. Cloud Comput. CSCloud 2017 3rd IEEE Int. Conf. Scalable Smart Cloud, SSC 2017 (August) (2017) 97–103. doi:10.1109/CSCloud.2017.15.

[85] P. Khilar, V. Chaudhari, R. Swain, Trust-Based Access Control in Cloud Computing Using Machine Learning: Intelligent Edge, Fog and Mist Computing, 2019, pp. 55–79. doi:10.1007/978-3-030-03359-0_3.

[86] J. Wang, S. Hu, Q. Wang, Y. Ma, Privacy-preserving outsourced feature extractions in the cloud: A survey, IEEE Netw. 31 (5) (2017) 36–41. doi:10.1109/MNET.2017.1600240.

[87] S. Bouchenak, G. Chockler, H. Chockler, G. Gheorghe, N. Santos, A. Shraer, Verifying cloud services, ACM SIGOPS Oper. Syst. Rev. 47 (2) (2013) 6–19. doi:10.1145/2506164.2506167.

[88] G. Grispos, T. Storer, W. B. Glisson, Calm Before the Storm: The Challenges of Cloud Computing in Digital Forensics, Int. J. Digit. Crime Forensics 4 (2) (2014) 28–48. arXiv:1410.2123, doi:10.4018/jdcf.2012040103.

[89] A. Aminnezhad, A. Dehghantanha, M. Taufik Abdullah, M. Damshenas, Cloud Forensics Issues and Opportunities, Int. J. Inf. Process. Manag. 4 (4) (2013) 76–85. doi:10.4156/ijipm.vol4.issue4.9.

[90] J. Dykstra, A. T. Sherman, Acquiring forensic evidence from infrastructure-as-a-service cloud computing: Exploring and evaluating tools, trust, and techniques, Digit. Investig. 9 (SUPPL.) (2012). doi:10.1016/j.diin.2012.05.001.

[91] C. Cadar, D. Dunbar, D. Engler, Klee: Unassisted and automatic generation of high-coverage tests for complex systems programs, in: Proc. 8th USENIX Symp. Oper. Syst. Des. Implementation, OSDI 2008, 2019, pp. 209–224.

[92] J. C. King, A new approach to program testing, in: Proc. 1975 Int. Conf. Reliab. Softw., Association for Computing Machinery, Inc, New York, New York, USA, 1975, pp. 228–233. doi:10.1145/800027.808444.

[93] V. Chipounov, V. Kuznetsov, G. Candea, The S2E platform: Design, implementation, and applications, in: ACM Trans. Comput. Syst., Vol. 30, 2012, pp. 1–49. doi:10.1145/2110356.2110358.

[94] J. Fridrich, J. Kodovsky, Rich models for steganalysis of digital images, IEEE Trans. Inf. Forensics Secur. 7 (3) (2012) 868–882. doi:10.1109/TIFS.2012.2190402.

[95] X. Han, L. S. Davis, Learning Rich Features for Image Manipulation Detection RPN layer RGB stream input RGB RoI features Bilinear Noise stream input Noise Conv Layers Noise RoI features, Cvpr (2018) 1313–1328

[96] D. Cozzolino, G. Poggi, L. Verdoliva, Splicebuster: A new blind image splicing detector, in: 2015 IEEE Int. Work. Inf. Forensics Secur. WIFS 2015 - Proc., IEEE, 2015, pp. 1–6. doi:10.1109/WIFS.2015.7368565.

[97] Y. Rao, J. Ni, A deep learning approach to detection of splicing and copy-move forgeries in images, in: 8th IEEE Int. Work. Inf. Forensics Secur. WIFS 2016, IEEE, 2017, pp. 1–6. doi:10.1109/WIFS.2016.7823911.

[98] R. Zhang, F. Zhu, J. Liu, G. Liu, Efficient feature learning and multi-size image steganalysis based on CNN, Preprint (2) (2018) 1–10. arXiv:1807.11428.
URL http://arxiv.org/abs/1807.11428

[99] J. Kodovský, J. Fridrich, V. Holub, Ensemble classifiers for steganalysis of digital media, in: IEEE Trans. Inf. Forensics Secur., Vol. 7, 2012, pp. 432–444. doi:10.1109/TIFS.2011.2175919.

[100] M. Saleem, M. R. Warsi, S. Islam, Learning rich features from software-as-a-service cloud computing for detecting trust violations, in: Lect. Notes Networks Syst., Vol. 116, 2020, pp. 445–452. doi:10.1007/978-981-15-3020-3_38.

[101] M. Saleem, M. R. Warsi, S. Islam, Feature Evaluation for Learning Underlying Data-Processing to Enhance Cloud Trust Through Rich Models, in: ICT Compet. Strateg., 2020, pp. 539–544. doi:10.1201/9781003052098-56.

[102] J. Li, G. Wu, Image recapture detection through residual-based local descriptors and machine learning, Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) 10603 LNCS (2017) 653–660. doi:10.1007/978-3-319-68542-7_56.

[103] D. Cozzolino, G. Poggi, L. Verdoliva, Recasting residual-based local descriptors as convolutional neural networks: An application to image forgery detection, in: IH MMSec 2017 - Proc. 2017 ACM Work. Inf. Hiding Multimed. Secur., 2017, pp. 159–164. arXiv:1703.04615, doi:10.1145/3082031.3083247.