Security and Privacy in Cloud-Based Big Data Systems: A Layered Risk Framework and Research Agenda

Kavitha Samala
ETL Quality Assurance Lead, Ven Soft LLC, Piscataway, NJ, USA

Abstract

Big data systems and cloud platforms are now deployed together. Elastic storage and compute make large-scale analytics practical, and they also concentrate sensitive data in shared, remotely managed infrastructure. This paper studies the security and privacy consequences of that concentration. It reviews the big data lifecycle and the NIST cloud model, then synthesizes published threat analyses, privacy principles, and service-model differences into a Layered Cloud Big Data Risk Framework. The framework maps risks and controls to five layers: identity and access, data and metadata, processing, platform service, and governance. A structured comparison of Infrastructure as a Service, Platform as a Service, and Software as a Service shows how operational responsibility shifts while tenant duties for classification, purpose, and retention remain. The paper states research propositions on misconfiguration, data minimization, cross-border processing, and auditability. The contribution is an analytical framework and agenda for later empirical testing, grounded in cited standards and surveys.

Keywords: big data, cloud computing, security, privacy, governance, risk framework, service models

1. Introduction

Cloud computing provides on-demand access to a shared pool of configurable resources with rapid elasticity and measured service (Mell & Grance, 2011). Big data workloads use that model because volume, velocity, and variety exceed the assumptions of a single server and a fixed schema (Laney, 2001; Chen et al., 2014). The combination supports analytics that many organizations could not host alone (Armbrust et al., 2010; Manyika et al., 2011). It also changes the security problem. Data are replicated across commodity machines, processed by short-lived clusters, queried by many roles, and often stored in a provider's region under the provider's administrative authority.

Surveys of big data on clouds list privacy, heterogeneity, and data movement among the open issues (Hashem et al., 2015; Assunção et al., 2015). Security surveys of cloud service delivery describe risks that differ across infrastructure, platform, and software services (Subashini & Kavitha, 2011; Fernandes et al., 2014). Guidance from NIST treats security and privacy in public clouds as a shared undertaking, with distinct provider and customer duties (Jansen & Grance, 2011). Data-protection law adds purpose limitation, minimization, and rights of access and erasure when personal data are involved (European Parliament and Council, 2016).

What is still needed is a single account that ties those strands to the big data lifecycle. A control that is sufficient for a virtual machine may be weak for a data lake in which one policy covers years of raw events. A contract that names a region may be incomplete if a transient processing job copies a partition elsewhere. This paper addresses that gap with a layered risk framework and a comparison of service models.

The research questions are:

1. Which security and privacy risks attach to each stage of cloud-based big data processing?
2. How do those risks and the corresponding controls differ across infrastructure, platform, and software service models?
3. Which research propositions follow for evaluation in later empirical work?

Section 2 presents the conceptual basis. Section 3 describes the method. Section 4 develops the framework. Section 5 compares service models. Section 6 states propositions and an agenda. Section 7 concludes.

2. Conceptual Basis

2.1 Big data lifecycle on cloud infrastructure

A practical lifecycle has five stages. Ingest accepts events and files. Store retains raw and curated data, usually on replicated distributed storage or object storage (Ghemawat et al., 2003; Dean & Ghemawat, 2008). Process runs batch, micro-batch, or stream jobs, often with an engine such as Spark that recomputes lost partitions (Zaharia et al., 2010; Zaharia et al., 2016). Serve exposes aggregates to applications and analysts. Govern records owners, schemas, retention, and access. NIST's big data framework offers neutral names for these roles, which helps when vendor products use different labels for similar functions (NIST Big Data Public Working Group, 2019).

Architectural choices at each stage affect risk. Replication improves availability and multiplies copies that must be protected. Elastic workers improve cost and create short-lived identities that must be issued and revoked. A durable log improves replay and becomes a high-value target if payloads are personal (Kreps et al., 2011). Separation of storage and compute, which is a main economic benefit of cloud analytics, means that authorization has to travel with the data rather than with a single cluster perimeter.

2.2 Cloud characteristics that shape the threat surface

NIST's essential characteristics explain why traditional perimeter controls are incomplete (Mell & Grance, 2011). On-demand self-service lets teams create stores and clusters quickly, including stores with weak defaults. Broad network access exposes management and query interfaces to the network. Resource pooling places tenant workloads on shared infrastructure, so isolation is a provider and configuration property. Rapid elasticity creates and destroys compute while data remain. Measured service creates billing and telemetry streams that can themselves reveal behavior.

Deployment models change the boundary of control. A private cloud keeps infrastructure inside one organization. A public cloud places it with a provider. A hybrid cloud splits stages across those boundaries. Buyya et al. (2009) and Zhang et al. (2010) describe this utility model and the research challenges of provisioning, energy, and security that come with it. For big data, hybrid designs are often motivated by data location: raw personal data stay in one environment, and elastic processing runs in another. That split is only as strong as the transfer policy that governs it.

2.3 Security, privacy, and management challenges in the literature

Takabi et al. (2010) organize cloud security and privacy challenges around identity management, access control, policy integration, and trust. Zissis and Lekkas (2012) emphasize confidentiality, integrity, and availability, and they highlight multi-tenancy and the dependence on the provider's administrative staff. Subashini and Kavitha (2011) show that the issues change with the service model: infrastructure customers face guest and network configuration, while software customers depend more on the provider's application controls and on isolation among tenants. Fernandes et al. (2014) consolidate a wide threat set, including insecure interfaces, account compromise, and data leakage.

The Cloud Security Alliance's threat reporting has repeatedly identified data breaches, weak credentials, insecure interfaces, and misconfiguration as leading issues in cloud environments (Cloud Security Alliance, 2013). Those issues scale with big data because a single broad grant on a raw zone can expose far more history than a breach of one operational table. Sivarajah et al. (2017) add that big data programs fail for process and management reasons as well as technical ones. Security programs that ignore ownership, quality, and retention fall into that class.

Privacy is related and distinct. Security controls reduce unauthorized access. Privacy principles constrain authorized processing: why data are collected, how little is kept, how long they remain, and which rights people can exercise (European Parliament and Council, 2016). Jansen and Grance (2011) advise public-cloud customers to assess data sensitivity, isolation, and residual risk before migration. A big data platform that retains raw events for possible later use conflicts with minimization even when encryption and identity controls are in place.

2.4 Consistency, distribution, and integrity

Distributed stores make explicit tradeoffs among consistency, availability, and partition tolerance (Brewer, 2000; Gilbert & Lynch, 2002). Analytics systems often accept a delay or a replay step so that ingest can continue during partial failure (Cattell, 2011; Marz & Warren, 2015). Integrity for security purposes is a related requirement: readers should see a committed snapshot, and a retried job should not silently duplicate personal records. Transactional metadata on cloud object storage is one published response to atomic publication of table versions (Armbrust et al., 2020). The risk framework in this paper treats integrity of analytical outputs as a security property, alongside confidentiality and availability.

3. Method

This is a structured analytical study. The sources are standards, threat reports, and peer-reviewed surveys and systems papers that are widely used in the field. The procedure has four steps.

First, the lifecycle in Section 2.1 is fixed as the unit of analysis, so risks are attached to stages rather than to product names. Second, threats and duties are extracted from the security, privacy, and big data cloud literature cited above and grouped by the asset they affect: identities, data, jobs, provider platforms, and organizational controls. Third, each group is stated as a layer with a risk statement, a tenant duty, and a provider duty. Fourth, the layers are applied to IaaS, PaaS, and SaaS using NIST's service definitions (Mell & Grance, 2011). The comparison asks, for each layer, where configuration authority sits.

The study produces a framework and propositions. It produces measured breach rates or control effectiveness scores only if a later study collects them. Where the literature supports a directional claim, the paper reports that claim with a citation. Where evidence is architectural, the paper says so.

4. Layered Cloud Big Data Risk Framework

The Layered Cloud Big Data Risk Framework has five layers. Each layer states the asset, the typical failure, the tenant duty, and the provider duty. Controls at a lower layer remain necessary when a higher layer is well designed.

4.1 Identity and access layer

The asset is every principal that can read or write data: people, pipeline jobs, and administrators. The typical failure is an over-broad, long-lived credential, or an account that survives after a project ends. Takabi et al. (2010) treat identity and access control as central cloud challenges because tenants and providers both issue identities, and policies must be integrated across them.

Tenant duties are least privilege, separation of ingest, processing, and analyst roles, time-bound human access, and unique identities for jobs. Provider duties are strong authentication for provider staff, protection of the identity service, and logs of administrative access. In a big data setting, least privilege is defined on tables, partitions, and columns. A role that can launch a cluster is different from a role that can read the raw zone.

4.2 Data and metadata layer

The asset is raw files, curated tables, logs, catalogs, and backups. The typical failure is an unauthorized copy, an unencrypted store, a missing retention rule, or a catalog that reveals sensitive combinations even when payloads are restricted. Replication and backups, which availability requires, are part of this asset (Ghemawat et al., 2003).

Tenant duties are classification, encryption decisions, key ownership where the threat model requires it, minimization of curated fields, and retention that deletes expired partitions. Provider duties are durability of storage, isolation between tenants, and protection of the storage service. Jansen and Grance (2011) emphasize assessing sensitivity before data are placed in a public cloud. For analytics, classification belongs on the data set and on the purpose, because the same column can be acceptable in an aggregate and inappropriate in a raw extract.

Metadata deserves explicit treatment. Schemas, partition names, and query histories can expose health, location, or financial categories. Catalog security is therefore part of the data layer, not an administrative afterthought.

4.3 Processing layer

The asset is jobs, intermediate files, shuffle data, model features, and serving snapshots. The typical failure is a job that reads more than its purpose allows, writes intermediate personal data to a world-readable temporary path, or publishes a result that cannot be traced to inputs. Spark-style lineage aids recomputation after failure (Zaharia et al., 2010). Lineage is also a privacy artifact: it describes how personal data were derived, and it should be available to authorized auditors and withheld from other tenants.

Tenant duties are purpose-scoped jobs, quarantine of malformed records, idempotent publishes, and a preference for aggregates over raw extracts in the serving zone. Provider duties are worker isolation, protection of temporary disks, and assured wiping when a virtual machine or container is released. Elastic compute makes wiping a routine event. A process that snapshots disks for debugging can undo that wiping if snapshots are retained without the same policy as the source data.

4.4 Platform service layer

The asset is the provider's virtualization, network, object store, managed analytics engine, and support channel. The typical failure is a vulnerable interface, weak tenant isolation, or a support process that grants broad access during an incident (Subashini & Kavitha, 2011; Cloud Security Alliance, 2013). Customers cannot patch the hypervisor or the provider's control plane. They can choose services, regions, and configurations, and they can require evidence of the provider's controls.

Provider duties include physical security, logical isolation, vulnerability management of the platform, and a documented way to report incidents. Tenant duties include selecting regions, disabling unused interfaces, and refusing features that replicate data outside the approved boundary. Zhang et al. (2010) note that security remains a research challenge in cloud platforms. For big data services, the managed query engine and the object store are on the critical path, because most analytical access bypasses a traditional application server.

4.5 Governance layer

The asset is decisions: purposes, owners, lawful basis, vendor terms, retention schedules, and evidence that controls operate. Sivarajah et al. (2017) show that management challenges sit beside data and process challenges in big data programs. A technically sound lake without an owner for each table fails this layer.

Tenant duties are an inventory of data sets, a named owner, a purpose statement, a retention period, a transfer record, and a review of provider contracts against that inventory. Provider duties are transparency about subprocessors, regions, and incident notification, to the extent the contract and applicable law require. The General Data Protection Regulation's principles of lawfulness, purpose limitation, minimization, and storage limitation give this layer a concrete test when personal data are processed (European Parliament and Council, 2016). Other jurisdictions impose their own duties. The framework does not replace legal advice. It requires that the technical design be able to carry out the duties the organization has accepted.

4.6 How the layers interact

A failure usually crosses layers. An analyst credential from layer 1, granted on a raw bucket in layer 2, used from an ad hoc job in layer 3, on a public service in layer 4, with no owner review in layer 5, is one incident with five gaps. Controls compose in the other direction. Governance assigns the purpose. The processing layer implements it. The data layer limits what the job can touch. Identity limits who can submit the job. The platform isolates that activity from other tenants. An assessment that scores only encryption, or only the provider's certification, describes one layer and leaves the others unexamined.

5. Comparison across Service Models

NIST distinguishes infrastructure, platform, and software services by the layer the customer controls (Mell & Grance, 2011). The framework compares them by asking who can change each control. The entries below are analytical judgments from that definition and from service-model security surveys (Subashini & Kavitha, 2011; Fernandes et al., 2014). They are a comparison of responsibility, and they are a ranking of measured risk.

5.1 Infrastructure as a Service

The tenant controls the guest system, the cluster layout, the analytical engine, network rules, and data placement on attached or object storage. That control supports custom retention, custom isolation, and placement of raw data in a chosen region. It also places guest hardening, key handling, and job identities with the tenant. Misconfiguration risk is high because self-service can expose a storage container or a master node. Provider duties concentrate on physical infrastructure, virtualization, and the base network (Zissis & Lekkas, 2012; Jansen & Grance, 2011).

IaaS fits workloads with strict placement rules and teams able to operate distributed systems. Big data on IaaS inherits the full processing-layer duty: temporary files, compaction jobs, and debug snapshots are the tenant's to secure.

5.2 Platform as a Service and managed analytics

The provider operates the engine, scaling, and much of patching. The tenant configures data sets, identities, job code, and often network and key options. Time to deploy falls, which is aligned with elasticity as described by Armbrust et al. (2010). The tenant's residual risk moves toward configuration of the managed service: public access flags, overly broad roles, region settings, and logging that was left disabled. Lock-in appears when authorization, metadata, or query features exist only inside one provider. Portability is a security concern as well as a commercial one, because exit must include deletion and transfer of all replicas the service created.

PaaS fits pipelines that can be expressed in a supported engine, with governance still owned by the tenant. The data and governance layers stay with the customer even when the provider patches the cluster.

5.3 Software as a Service

The provider operates the application and, in multi-tenant products, the analytical store behind it. The tenant administers users, sharing settings, and retention options that the product exposes. Subashini and Kavitha (2011) note the customer's dependence on provider application controls in this model. Big data risks appear as exports, integrated add-ons, and support access. A tenant who cannot see the underlying partitions must rely on contractual evidence, audit reports, and product settings. Minimization is possible when the product offers field suppression, aggregation, and deletion, and it is blocked when the product retains detailed histories the tenant cannot purge.

SaaS fits consumption of analytics by end users. It is a weak fit when the organization must prove a custom lineage or must run arbitrary jobs over raw personal data.

5.4 Summary comparison

Layer: Identity
IaaS tenant emphasis: Guest, job, and user identities
PaaS tenant emphasis: Service roles and user identities
SaaS tenant emphasis: User and sharing administration

Layer: Data
IaaS tenant emphasis: Placement, encryption, retention implementation
PaaS tenant emphasis: Bucket and table policy, keys, region
SaaS tenant emphasis: Settings the product exposes; contract for the rest

Layer: Processing
IaaS tenant emphasis: Engine, temporary data, job scope
PaaS tenant emphasis: Job code and purpose scope on a managed engine
SaaS tenant emphasis: Feature use; exports and integrations

Layer: Platform
IaaS tenant emphasis: Choice of provider controls and region
PaaS tenant emphasis: Choice of managed service options
SaaS tenant emphasis: Choice of vendor and enabled modules

Layer: Governance
IaaS tenant emphasis: Full inventory and operation of evidence
PaaS tenant emphasis: Inventory plus configuration evidence
SaaS tenant emphasis: Inventory, contract review, and product evidence

Across all three models, purpose, classification, and accountability stay with the tenant. The provider's share of technical operation grows from IaaS to SaaS. Hybrid deployments combine rows of this table. A raw zone on private IaaS and a serving dashboard on public SaaS is a coherent design when the governance layer records what may cross the boundary and the data layer enforces it.

5.5 Lifecycle view of priority risks

Stage: Ingest
Priority risk: Unauthenticated producers; excess payload
Control: Job identity; schema that rejects unneeded fields

Stage: Store
Priority risk: Open repositories; unmanaged replicas
Control: Classification, encryption, retention, replica inventory

Stage: Process
Priority risk: Over-broad reads; dirty temporary data
Control: Purpose-scoped jobs; isolated scratch space; wiping

Stage: Serve
Priority risk: Detailed extracts replacing aggregates
Control: Serving zone of minimized tables; column limits

Stage: Govern
Priority risk: Unknown owners; transfers without a record
Control: Catalog ownership; contract and region register

This table answers the first research question. Risk is stage-specific, and the control sits in a named layer so that an audit can test it.

6. Propositions and Research Agenda

The framework implies propositions that a later study can test. They are stated so that confirmation and disconfirmation are both possible.

P1. Misconfiguration dominates tenant-controlled exposure. In IaaS and PaaS big data deployments, incidents that the tenant could have prevented will more often trace to access policy, public endpoints, and logging gaps than to a novel cryptographic break. This proposition is consistent with the emphasis on credentials, interfaces, and misconfiguration in cloud threat reporting (Cloud Security Alliance, 2013). A test would classify incidents by layer.

P2. Minimization reduces impact more than collection volume alone would predict. Platforms that publish aggregates and drop unused raw fields will show a smaller exposed surface under the same identity failure than platforms that grant analysts direct raw access. The mechanism is the data and

processing layers. A test would compare accessible field counts and retention under two catalog policies.

P3. Hybrid placement improves sovereignty only when transfer rules are enforced in the data path. A documented architecture that keeps sensitive raw data in one region, while jobs can still read those buckets from another region, does not meet the governance claim. A test would attempt a cross-boundary read and record whether policy blocks it.

P4. Auditability depends on lineage and administrative logs being retained under the same access discipline as the data. If pipeline logs and query histories are world-readable inside the tenant, the governance layer's evidence becomes a second store of sensitive information. A test would review permissions on logs as well as on tables.

P5. Service model shifts labor and changes the evidence available to the tenant. Moving from IaaS to SaaS reduces guest-hardening work and increases reliance on provider reports and product settings (Subashini & Kavitha, 2011). A test would measure time spent on patching versus time spent on contract and configuration review. It would also record which controls the tenant can observe directly.

A workable empirical design is a multi-case study of analytics platforms, scoring each layer from documents and configuration, plus a tabletop exercise of P3. Quantitative breach studies need incident data the present paper does not have. Publishing the scoring guide would let other researchers repeat the assessment.

6.1 Implications for practice

Teams can use the five layers as a review checklist before a data set is onboarded. The review names the owner, the purpose, the region, the principals, the retention, and the serving grain. Pipelines then implement those decisions. Provider certification supports the platform layer and does not close the identity, data, or governance layers. Exit plans belong in the initial design: where replicas exist, how deletion is demonstrated, and which metadata must leave with the data.

6.2 Limitations

The framework is synthesized from published standards and research. It has not been validated here against a set of incident reports collected for this study. Service-model comparisons describe responsibility, and they describe a measured frequency of harm. Legal duties differ by jurisdiction. The GDPR is used as a clear, widely studied statement of privacy principles, and organizations must apply the law that actually governs them. Threat technique details are omitted on purpose. The paper names failure classes and duties so that defenders can assess systems, and it withholds procedures that would help misuse a cloud environment.

7. Conclusion

Cloud-based big data systems create value by pooling elastic compute and large stores, and they create risk by multiplying copies, principals, and regions. This paper answered three questions. Risks attach differently to ingest, storage, processing, serving, and governance, with priority failures in credentials, open stores, over-broad jobs, detailed extracts, and missing ownership. IaaS, PaaS, and SaaS move technical operation toward the provider in that order, while classification, purpose, and accountability remain tenant duties. Five propositions follow, centered on misconfiguration, minimization, enforceable transfer rules, protected audit trails, and the change in evidence as managed services are adopted.

The Layered Cloud Big Data Risk Framework is the contribution that ties lifecycle stages to those duties. It gives implementers a review structure and gives researchers variables for case studies. Measured tests of the propositions are the next step. They belong in a study that inspects real configurations and records what it finds.

References

1) Armbrust, M., Das, T., Sun, L., Yavuz, B., Zhu, S., Murthy, M., Torres, J., van Hovell, H., Ionescu, A., Łuszczak, A., Świtakowski, M., Szafrański, M., Li, X., Ueshin, T., Mokhtar, M., Boncz, P., Ghodsi, A., Paranjpye, S., Senster, P., Xin, R., & Zaharia, M. (2020). Delta Lake: High-performance ACID table storage over cloud object stores. Proceedings of the VLDB Endowment, 13(12), 3411–3424.

2) Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., & Zaharia, M. (2010). A view of cloud computing. Communications of the ACM, 53(4), 50–58. https://doi.org/10.1145/1721654.1721672

3) Assunção, M. D., Calheiros, R. N., Bianchi, S., Netto, M. A. S., & Buyya, R. (2015). Big data computing and clouds: Trends and future directions. Journal of Parallel and Distributed Computing, 79–80, 3–15. https://doi.org/10.1016/j.jpdc.2014.08.003

4) Brewer, E. A. (2000). Towards robust distributed systems. In Proceedings of the Nineteenth Annual ACM Symposium on Principles of Distributed Computing. ACM. https://doi.org/10.1145/343477.343502

5) Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation Computer Systems, 25(6), 599–616. https://doi.org/10.1016/j.future.2008.12.001

6) Cattell, R. (2011). Scalable SQL and NoSQL data stores. ACM SIGMOD Record, 39(4), 12–27. https://doi.org/10.1145/1978915.1978919

7) Chen, M., Mao, S., & Liu, Y. (2014). Big data: A survey. Mobile Networks and Applications, 19(2), 171–209. https://doi.org/10.1007/s11036-013-0489-0

8) Cloud Security Alliance. (2013). The notorious nine: Cloud computing top threats in 2013. Cloud Security Alliance.

9) Dean, J., & Ghemawat, S. (2008). MapReduce: Simplified data processing on large clusters. Communications of the ACM, 51(1), 107–113. https://doi.org/10.1145/1327452.1327492

10) European Parliament and Council. (2016). Regulation (EU) 2016/679 (General Data Protection Regulation). Official Journal of the European Union, L119, 1–88.

11) Fernandes, D. A. B., Soares, L. F. B., Gomes, J. V., Freire, M. M., & Inácio, P. R. M. (2014). Security issues in cloud environments: A survey. International Journal of Information Security, 13(2), 113–170. https://doi.org/10.1007/s10207-013-0208-7

12) Ghemawat, S., Gobioff, H., & Leung, S.-T. (2003). The Google file system. In Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles (pp. 29–43). ACM. https://doi.org/10.1145/945445.945450

13) Gilbert, S., & Lynch, N. (2002). Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. ACM SIGACT News, 33(2), 51–59. https://doi.org/10.1145/564585.564601

14) Hashem, I. A. T., Yaqoob, I., Anuar, N. B., Mokhtar, S., Gani, A., & Ullah Khan, S. (2015). The rise of "big data" on cloud computing: Review and open research issues. Information Systems, 47, 98–115. https://doi.org/10.1016/j.is.2014.07.006

15) Jansen, W., & Grance, T. (2011). Guidelines on security and privacy in public cloud computing (NIST Special Publication 800-144). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-144

16) Kreps, J., Narkhede, N., & Rao, J. (2011). Kafka: A distributed messaging system for log processing. In Proceedings of the NetDB Workshop.

17) Laney, D. (2001). 3D data management: Controlling data volume, velocity, and variety (Application Delivery Strategies, File 949). META Group.

18) Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., & Hung Byers, A. (2011). Big data: The next frontier for innovation, competition, and productivity. McKinsey Global Institute.

19) Marz, N., & Warren, J. (2015). Big data: Principles and best practices of scalable realtime data systems. Manning.

20) Mell, P., & Grance, T. (2011). The NIST definition of cloud computing (NIST Special Publication 800-145). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-145

21) NIST Big Data Public Working Group. (2019). NIST big data interoperability framework (NIST Special Publication 1500). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.1500-1r2

22) Sivarajah, U., Kamal, M. M., Irani, Z., & Weerakkody, V. (2017). Critical analysis of big data challenges and analytical methods. Journal of Business Research, 70, 263–286. https://doi.org/10.1016/j.jbusres.2016.08.001

23) Subashini, S., & Kavitha, V. (2011). A survey on security issues in service delivery models of cloud computing. Journal of Network and Computer Applications, 34(1), 1–11. https://doi.org/10.1016/j.jnca.2010.07.006

24) Takabi, H., Joshi, J. B. D., & Ahn, G.-J. (2010). Security and privacy challenges in cloud computing environments. IEEE Security & Privacy, 8(6), 24–31. https://doi.org/10.1109/MSP.2010.186

25) Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., & Stoica, I. (2010). Spark: Cluster computing with working sets. In Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing. USENIX Association.

26) Zaharia, M., Xin, R. S., Wendell, P., Das, T., Armbrust, M., Dave, A., Meng, X., Rosen, J., Venkataraman, S., Franklin, M. J., Ghodsi, A., Gonzalez, J., Shenker, S., & Stoica, I. (2016). Apache Spark: A unified engine for big data processing. Communications of the ACM, 59(11), 56–65. https://doi.org/10.1145/2934664

27) Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing: State-of-the-art and research challenges. Journal of Internet Services and Applications, 1(1), 7–18. https://doi.org/10.1007/s13174-010-0007-6

28) Zissis, D., & Lekkas, D. (2012). Addressing cloud computing security issues. Future Generation Computer Systems, 28(3), 583–592. https://doi.org/10.1016/j.future.2010.12.006