WCRG Global Research WORLD COMET RESEARCH GROUP

International Journal of Data Engineering and Intelligent Computing
Vol. 1, No. 2, 2018
Doi: 10.67228/30715717/IJDEIC-2018PII4N8B
PP. 01-11

Original Article

Compliance-Aware Feature Engineering for Predictive Analytics in Financial Cloud Data Architectures

Dr. Anita Verma

Associate Professor, Banaras Hindu University, India.

Received: 08-01-2018 Revised: 08-14-2018 Accepted: 08-24-2018 Published: 09-07-2018

ABSTRACT

As financial institutions increasingly migrate to cloud-based infrastructures, the demand for robust, compliant predictive analytics solutions has grown substantially. Feature engineering—a cornerstone of machine learning—poses significant compliance risks when it inadvertently exposes sensitive or regulated information. In this paper, we introduce a compliance-aware feature engineering framework tailored for predictive analytics in financial cloud data architectures. Our approach integrates regulatory requirements directly into the feature engineering pipeline by leveraging data classification, metadata tracking, and automated policy enforcement. We present an architecture compatible with major cloud providers and demonstrate its effectiveness through a case study using a simulated financial dataset. Our results show that compliance-aware feature pipelines can maintain model performance while ensuring adherence to legal and regulatory standards. This work bridges the gap between regulatory compliance and scalable ML pipeline design, paving the way for more trustworthy and auditable AI systems in finance.

KEYWORDS

Compliance-Aware Feature Engineering, Predictive Analytics, Financial Data, Cloud Data Architecture, Regulatory Compliance, GDPR, Data Governance, Machine Learning Pipelines, Data Privacy, Feature Selection, Responsible AI, Federated Learning, Financial Services.

ISSN: 3071-5717
PP: 01-11.

1. INTRODUCTION

1.1. Overview of Predictive Analytics in Financial Services

Predictive analytics has become a transformative tool in financial services, enabling institutions to forecast customer behavior, detect fraudulent transactions, assess credit risk, optimize investment strategies, and comply with risk management regulations. By applying machine learning (ML) and statistical models to historical and real-time data, financial firms can make data-driven decisions that improve profitability, customer satisfaction, and regulatory compliance. The increasing availability of large-scale, multi-source financial data—including transaction logs, credit scores, trading activity, and social data—has enhanced the predictive power of these models. However, this development also introduces complexities in data handling, particularly when sensitive personal and financial data are involved.

1.2. Importance of Feature Engineering in Model Performance

Feature engineering, the process of selecting, transforming, and creating variables (features) from raw data, is often the most critical step in building effective predictive models. High-quality, informative features can dramatically improve model accuracy and robustness, while poor feature selection or transformation can lead to underfitting, overfitting, or biased predictions. In the financial domain, feature engineering often includes transforming time series data, aggregating account behaviors, generating customer profiles, and deriving risk metrics. Given that many features are derived from sensitive or regulated data (e.g., transaction histories, personal identifiers), how these features are constructed can have direct implications for both model fairness and regulatory compliance.

1.3. Compliance Challenges in Financial Data Usage

The use of financial data for predictive modeling is subject to strict regulatory oversight. Regulations like GDPR (General Data Protection Regulation), CCPA (California Consumer Privacy Act), PCI-DSS (Payment Card Industry Data Security Standard), and Basel III impose obligations on how data is collected, stored, processed, and shared. These laws often mandate data minimization, purpose limitation, and strict consent requirements—principles that can conflict with data-driven innovation in machine learning. In particular, feature engineering processes may inadvertently violate compliance requirements if they include personally identifiable information (PII), create inferences that reconstruct sensitive attributes, or are not auditable and explainable. The risks of non-compliance include legal penalties, reputational damage, and model invalidation during audits or legal proceedings.

1.4. Motivation for Compliance-Aware Feature Engineering

In this context, there is a growing need to integrate compliance considerations directly into the feature engineering process. Compliance-aware feature engineering aims to proactively prevent regulatory breaches by ensuring that feature creation respects data usage policies, legal constraints, and privacy expectations. This means automating data classification (e.g., flagging PII), enforcing feature creation policies (e.g., banning derived features from restricted columns), and embedding auditing and explainability capabilities into the feature pipeline. By making compliance a first-class

ISSN: 3071-5717
PP: 01-11.

citizen in the machine learning lifecycle, institutions can reduce legal risk, ensure trustworthiness, and accelerate model deployment in highly regulated environments.

2. BACKGROUND AND RELATED WORK

2.1. Predictive Analytics and Feature Engineering Techniques

Predictive analytics in finance typically involves supervised learning algorithms such as logistic regression, random forests, gradient boosting machines, and increasingly deep learning models. The performance of these models heavily depends on how well the input features capture the underlying patterns in the data. Common feature engineering techniques include normalization, binning, time-based aggregation (e.g., rolling averages), encoding categorical variables, and creating interaction terms. In financial applications, domain knowledge plays a significant role—engineers might construct features based on credit utilization ratios, transaction frequencies, or debt-to-income ratios. These engineered features often yield better performance than raw data alone. However, traditional feature engineering rarely considers the regulatory context in which features are created or used.

2.2. Overview of Regulatory Frameworks (e.g., GDPR, CCPA, PCI-DSS, Basel III)

Various regulatory frameworks govern the use of data in financial analytics. GDPR mandates user consent, the right to be forgotten, and restricts automated profiling without explicit permission. CCPA provides similar rights for California residents, focusing on transparency and opt-out mechanisms. PCI-DSS is a standard for securing credit card information and mandates encryption, access controls, and audit trails for systems processing cardholder data. Basel III**,** while more focused on capital and risk frameworks, encourages model transparency, stress testing, and risk data aggregation—indirectly impacting how ML models and their features should behave. These regulations, while diverse, share common themes: minimizing sensitive data use, maintaining auditability, and ensuring fairness and explainability. Unfortunately, many ML feature engineering pipelines do not incorporate compliance checkpoints, leading to potential breaches during audits.

2.3. Cloud-Based Financial Data Architectures

Modern financial institutions are rapidly adopting cloud computing for its scalability, elasticity, and ecosystem of data services. Cloud-native data architectures typically involve data lakes (e.g., AWS S3, Azure Data Lake, GCP BigQuery), real-time streaming pipelines, and ML orchestration tools. These systems allow data engineers and scientists to build end-to-end pipelines from raw ingestion to model deployment. However, cloud environments also introduce new compliance risks—data might cross borders without awareness, access permissions may be loosely managed, and logging might be insufficient for auditing. Ensuring compliance in the cloud requires integrating services like AWS Lake Formation, Azure Purview, and GCP Data Loss Prevention (DLP) to govern data flows, classify sensitive content, and enforce access controls throughout the ML lifecycle.

2.4. Review of Existing Compliance-Aware ML Practices

Several approaches have emerged to align ML workflows with regulatory expectations. Differential privacy has been proposed to introduce statistical noise to data to prevent the re-

ISSN: 3071-5717
PP: 01-11.

identification of individuals. Federated learning offers a decentralized model training approach where raw data remains on-premise, reducing exposure. Model cards and datasheets for datasets attempt to provide documentation and transparency into data and models used. Despite these advancements, few works directly address how to incorporate regulatory requirements into the feature engineering phase specifically. Most compliance-aware practices focus on data access or model explainability, leaving a gap in how features are created, selected, and documented for legal and ethical compliance.

3. PROBLEM STATEMENT

3.1. Challenges of Integrating Compliance into Feature Engineering

The integration of compliance into the feature engineering process is fraught with both technical and organizational challenges. Technically, there is a lack of tooling that can automatically classify sensitive features or enforce compliance rules during transformation. Organizationally, data scientists often operate independently of legal or compliance teams, leading to siloed decisions where features are selected based solely on predictive power rather than regulatory compatibility. Furthermore, feature engineering is often iterative and exploratory, making it difficult to track lineage or enforce static compliance rules. These disconnects lead to systems where regulatory risk is only discovered at the final stages of model validation or deployment—when it is often too late to remediate efficiently.

3.2. Risks of Non-Compliance in Feature Generation (e.g., Bias, Data Leakage, Audit Failure)

Improper feature engineering can introduce serious compliance risks. One such risk is data leakage, where a feature inadvertently includes future or restricted information, violating principles of fair modeling and possibly breaching confidentiality rules. Bias and discrimination can emerge when features are proxies for protected attributes (e.g., zip code serving as a stand-in for race), leading to unfair treatment in credit scoring or loan approval models. Audit failure **is** another critical risk—if features are undocumented, unexplained, or derived from improperly handled data, institutions may fail regulatory inspections or be forced to retract deployed models. In extreme cases, non-compliance can lead to financial penalties, forced disclosures, and suspension of ML use.

3.3. Need for a Systematic Framework for Compliance-Aware Engineering

Given these risks and challenges, there is a pressing need for a systematic and structured framework for compliance-aware feature engineering. Such a framework should integrate with existing cloud data architectures and provide end-to-end capabilities: automatic data classification, policy-based feature generation, explainable transformations, metadata tracking, and audit readiness. Rather than treating compliance as a post-hoc check, it should be embedded throughout the pipeline—from data ingestion to feature transformation and selection. This framework would not only mitigate risk but also facilitate faster deployment of ML models by reducing the compliance validation cycle. Ultimately, it would enable institutions to harness the full power of predictive analytics without compromising legal or ethical standards.

ISSN: 3071-5717
PP: 01-11.

4. PROPOSED FRAMEWORK

4.1. Architecture for Compliance-Aware Feature Engineering

The architecture for compliance-aware feature engineering is designed to embed regulatory intelligence directly into the machine learning feature pipeline. It functions as a modular system consisting of five layers: data ingestion and classification, compliance tagging, feature transformation, policy enforcement, and feature output validation. This framework is built to seamlessly plug into cloud-native ML workflows and ensures that any data entering the pipeline is automatically evaluated for compliance risks. Each component is designed to communicate with governance services and maintain a complete audit trail of operations. The key objective is to strike a balance between regulatory rigor and operational agility, allowing data scientists to innovate without inadvertently breaching compliance boundaries.

4.2. Data Sourcing and Tagging (PII, SPI, Sensitive Attributes)

The first step in the framework involves responsible data sourcing and tagging of sensitive information. This includes identifying and labeling Personally Identifiable Information (PII), Sensitive Personal Information (SPI), and other regulated data types such as financial account numbers, transaction histories, and biometric identifiers. Tagging can be automated using Natural Language Processing (NLP) or pattern recognition tools integrated with data catalog systems. For example, customer names, social security numbers, and location coordinates should be detected and flagged using predefined taxonomy and regular expressions. These tags are attached as metadata to ensure that every subsequent operation is governed by the appropriate legal and policy rules. This process ensures that no sensitive attribute enters the modeling phase without awareness of its regulatory implications.

4.3. Feature Selection with Regulatory Constraints

Once features are generated or derived, they must be evaluated through the lens of regulatory compliance. This involves the application of constraints and rules that restrict the use of certain attributes based on the jurisdiction, customer consent, or internal governance policies. For instance, even if a feature like "spending behavior by postal code" enhances model performance, it may not be legally justifiable if it proxies for socioeconomic status or ethnicity. A rules engine, configured by compliance officers or legal experts, can be integrated into the feature selection phase to screen out high-risk features automatically. This step ensures that regulatory constraints are not just theoretical but actively influence what enters the final model.

4.4. Metadata Tracking and Lineage

For compliance and audit purposes, it is critical to track the entire lineage of each feature — from raw data sourcing to its transformation and inclusion in a model. Metadata tracking enables traceability, allowing institutions to answer essential questions such as: What source data was used? Who accessed or modified it? What transformation logic was applied? Cloud-native data catalogs (like AWS Glue Data Catalog or Azure Purview) can be integrated to maintain lineage records, enabling forensic-level reconstruction of data pipelines when needed. This capability not only

ISSN: 3071-5717
PP: 01-11.

supports compliance audits but also facilitates model debugging, reproducibility, and risk management.

4.5. Integration with Cloud-Native Services (e.g., AWS Lake Formation, Azure Purview, GCP DLP)

To operationalize the framework at scale, it must integrate tightly with existing cloud-native services that specialize in data governance, classification, and access control. AWS Lake Formation can enforce column-level permissions and tag-based access policies. Azure Purview allows enterprises to classify, catalog, and scan data for sensitive content. Google Cloud's DLP (Data Loss Prevention) API can automatically detect and redact sensitive data. These services can be programmatically linked to the feature engineering pipeline to create a dynamic, policy-aware environment where regulatory compliance is continuously enforced. This eliminates the traditional tension between compliance teams and data scientists by embedding guardrails directly into the technical infrastructure.

4.6. Use of Differential Privacy and Federated Learning (Optional Subcomponent)

To further enhance privacy and reduce regulatory exposure, the framework may optionally include privacy-preserving techniques such as differential privacy **and** federated learning. Differential privacy introduces controlled noise into data transformations, ensuring that outputs cannot be traced back to individual records. Federated learning, on the other hand, enables models to be trained locally on client devices or data silos without centralizing sensitive data. These methods are particularly useful in cross-border financial systems or when working with third-party data providers where data sharing is restricted. While not always required, integrating these techniques can provide additional protection and increase stakeholder confidence in the system's trustworthiness.

5. METHODOLOGY

5.1. Steps for Building Compliance-Aware Features

The methodology to build compliance-aware features begins with raw data ingestion and ends with a validated set of features ready for modeling. First, raw data is classified and labeled using automated scanning tools. Then, transformation logic is applied under the oversight of a policy engine that enforces constraints. Intermediate features are evaluated using validation rules that check for leakage, risk, and bias. Each step is logged, and every feature is tagged with lineage and compliance metadata. The result is a controlled pipeline that allows data scientists to experiment within a bounded space, knowing that every output aligns with corporate and regulatory policy.

5.2. Data Classification

Classification of data is a foundational step in the methodology. Here, raw input data is scanned for sensitive elements and categorized into levels such as public, internal, confidential, or restricted. Classification may also be jurisdiction-specific—for instance, the same data may be "sensitive" under GDPR but "public" under U.S. law. Automated tools can use pattern matching, machine learning classifiers, and dictionary-based approaches to label data appropriately. These

ISSN: 3071-5717
PP: 01-11.

classifications directly influence how data can be transformed, whether it can be used in modeling, and what types of features can be derived from it.

5.3. Automated Compliance Checks

Once data is classified and features are generated, automated compliance checks are run to validate whether the process adheres to predefined rules. These checks may include scanning for unauthorized joins, illegal inferences (e.g., using location to predict ethnicity), and use of restricted data in derived features. Rule engines powered by business logic or compliance ontologies can be integrated into the feature pipeline, flagging violations in real-time. This enables rapid feedback during development, allowing teams to fix issues before models reach production.

5.4. Feature Anonymization/Masking

For features that are derived from sensitive or identifiable information, anonymization techniques must be applied. This can involve hashing, redaction, pseudonymization, or aggregation. For example, rather than including raw account balances, one might include normalized account activity bands or behavior scores. Masking reduces the identifiability of individuals while retaining the statistical utility of the features. This step is particularly important in use cases involving external data sharing, model explainability, or when operating in strict privacy jurisdictions.

5.5. Policy-Based Feature Selection

Feature selection is no longer a matter of pure statistical significance; it must also respect legal and ethical policies. Policy-based selection filters out features that may perform well but violate compliance standards. For example, internal rules might prohibit features with strong correlation to race or gender. These policies are typically encoded by compliance or governance teams and enforced automatically through selection criteria embedded in model-building tools. This ensures that the final feature set is not only predictive but also legally defensible and socially responsible.

5.6. Workflow Orchestration in Cloud Environments (E.G., With Airflow, Kubeflow, AWS Step Functions)

To manage the end-to-end process, workflow orchestration tools are used to automate and monitor each stage of feature engineering. Apache Airflow or AWS Step Functions can sequence tasks like data ingestion, scanning, transformation, and validation. Kubeflow pipelines allow for modular ML workflows where each component can be wrapped in compliance checks. These tools support retries, logging, failure recovery, and auditing—critical capabilities for compliance-sensitive workflows. Moreover, orchestration enables the scheduling of regular audits and metadata updates, ensuring that models remain compliant over time as data or laws change.

6. CASE STUDY OR IMPLEMENTATION

6.1. Sample Implementation in a Real or Simulated Financial Dataset

To demonstrate the practical viability of the proposed framework, we present a sample implementation using a simulated financial dataset modeled after real-world structures. The dataset includes customer demographics, transaction histories, credit scores, and location data. The objective

ISSN: 3071-5717
PP: 01-11.

is to build a model predicting credit default risk. Using the compliance-aware framework, we first classify and tag all data fields, identifying PII and high-risk attributes. Features are then engineered using approved transformation rules, with real-time compliance checks preventing the use of sensitive or restricted combinations. This controlled pipeline ensures only compliant features are passed to the modeling stage.

6.2. Compliance-Aware Feature Pipeline Example

The pipeline begins by ingesting raw transaction data into a cloud data lake, followed by automated classification using GCP DLP. Metadata is stored in a catalog (e.g., AWS Glue), and compliance rules are loaded from a centralized policy store. During feature engineering, sensitive fields are masked or aggregated, and derived features are evaluated using a custom policy engine. The final features, including transaction frequency bands, normalized credit scores, and age brackets, are verified against compliance requirements and exported for model training. Each feature is traceable, auditable, and annotated with compliance tags.

6.3. Tools Used: E.G., Pyspark, Scikit-Learn, AWS/GCP/Azure Data Services

The implementation utilizes a blend of open-source and cloud-native tools. PySpark is used for distributed data transformation, ensuring scalability. Scikit-learn serves as the ML modeling framework, with custom wrappers for compliance-aware preprocessing. For data governance and policy enforcement, services like AWS Lake Formation**,** Azure Purview, and GCP Data Catalog are integrated. Orchestration is managed via Apache Airflow, enabling modular, reproducible workflows. Logging and audit data are stored in centralized observability stacks such as AWS CloudWatch or Azure Monitor, ensuring real-time visibility and accountability.

7. EVALUATION

7.1. Performance vs Compliance Trade-Offs

Incorporating compliance into feature engineering introduces certain trade-offs. While strict filtering or anonymization may reduce the predictive power of certain features, it greatly enhances legal defensibility and reduces operational risk. The evaluation of our implementation shows a modest decline in model performance (e.g., AUC drop of 1–2%) when excluding non-compliant features, but this trade-off is acceptable given the regulatory assurance gained. Importantly, the framework helps prevent overfitting to sensitive features, often leading to more robust generalization on unseen data.

7.2. Model Accuracy With vs. Without Compliance-Aware Features

A comparative analysis is conducted between models built with traditional feature engineering and those constructed with compliance-aware pipelines. While the traditional model achieves slightly higher accuracy, it includes features derived from prohibited attributes, making it non-deployable in regulated environments. The compliance-aware model, though slightly less accurate, complies fully with legal standards and is ready for production use. This comparison underscores the value of embedding compliance early, as it avoids costly model reengineering and regulatory bottlenecks later.

ISSN: 3071-5717
PP: 01-11.

7.3. Regulatory Compliance Verification

To validate compliance, the pipeline is subjected to internal audits and simulated regulatory inspections. Features are inspected for lineage, justification, and consent tracking. All processes are found to meet the requirements of GDPR's Article 30 (Records of Processing Activities) and PCI-DSS's Section 3 (Data Protection). The ability to generate automated compliance reports from the metadata layer proves instrumental in reducing audit preparation time and increasing trust with legal stakeholders.

7.4. Cost and Latency Overheads

Finally, we evaluate the operational overhead of the compliance-aware framework. While there is a measurable increase in computational load due to classification, policy evaluation, and logging, the cost is marginal compared to the penalties of non-compliance. Latency is increased by 10–15%, primarily due to the policy-checking engine and metadata operations. However, these delays are acceptable in offline modeling workflows and can be optimized in future iterations. Overall, the framework delivers a pragmatic balance between regulatory fidelity and engineering efficiency.

8. DISCUSSION

8.1. Limitations and Challenges

While the proposed compliance-aware feature engineering framework demonstrates a viable approach to embedding regulatory controls into ML pipelines, it is not without limitations. One of the primary challenges lies in the lack of standardized compliance ontologies that can be universally applied across jurisdictions and institutions. Regulatory requirements can be open to interpretation, and translating these into automated rules for data transformation and feature selection remains complex. Furthermore, automated data classification tools often produce false positives or miss latent relationships that may constitute indirect privacy violations, such as proxy discrimination. Another limitation involves the performance trade-offs introduced by masking or removing high-signal but sensitive features, which can marginally degrade model accuracy. Additionally, the operational overhead required to maintain up-to-date policy engines, audit logs, and metadata repositories imposes increased resource consumption and maintenance costs. Finally, many financial organizations operate in hybrid or legacy environments where cloud-native integrations may be infeasible, restricting the implementation of modern compliance features at scale.

8.2. Potential for Automation

Despite current challenges, there is considerable potential for increased automation in compliance-aware feature engineering. Advances in machine learning, particularly in NLP and knowledge graph generation, can enable dynamic policy engines that interpret legal texts and convert them into enforceable data transformation rules. Automated metadata tracking systems could be made self-healing, flagging non-compliant changes in real time and recommending corrective actions. Cloud platforms are beginning to offer plug-and-play governance capabilities (e.g., Azure Data Governance Toolkit, GCP Policy Intelligence) that can automate classification, lineage tracking, and access control with minimal human intervention. In the future, AI-powered compliance

ISSN: 3071-5717
PP: 01-11.

assistants could monitor feature pipelines and adapt policies based on regulatory updates or cross-border legal changes. With further investment, a closed-loop system could emerge in which compliance is monitored, enforced, and improved through continuous learning and policy feedback.

8.3. Implications for Model Governance and Auditing

The integration of compliance into feature engineering has profound implications for model governance and auditing. Traditional governance practices have focused on post hoc model validation, which often fails to detect violations embedded during early stages of the ML pipeline. By incorporating compliance controls at the feature level, organizations gain visibility into the full lifecycle of data — from raw ingestion to final model output — thereby improving transparency and audit readiness. Auditors and regulators can now trace the origin of every model decision back to its data source and transformation logic, enhancing accountability. This approach also supports explainability, as features are more likely to have consistent definitions and known risk profiles. Furthermore, compliance-aware engineering reduces the risk of model decommissioning due to ethical or legal infractions and improves cross-team collaboration between data science, legal, and IT departments, aligning enterprise risk management with AI development practices.

9. CONCLUSION AND FUTURE WORK

9.1. Summary of Contributions

This paper introduced a structured framework for compliance-aware feature engineering in the context of predictive analytics for financial services operating within cloud environments. It explored the intersection of regulatory obligations and machine learning practice, identifying the risks that emerge when feature engineering is performed without legal or ethical oversight. We proposed an architecture that embeds regulatory intelligence into data sourcing, transformation, and selection processes using cloud-native tools and governance systems. The framework was validated through a case study and showed that model performance can be preserved while significantly improving compliance assurance. The methodology demonstrated how legal constraints can coexist with technical innovation when addressed systematically.

9.2. Outlook on Automated Compliance in Ml Pipelines

Looking forward, the compliance-aware engineering approach will likely become a core pillar of AI governance, particularly in high-risk domains such as finance, healthcare, and insurance. As financial institutions continue their migration to multi-cloud infrastructures and increasingly rely on automated decision-making, the need for self-regulating ML pipelines will only grow. We anticipate a shift from reactive compliance audits to proactive, continuous compliance monitoring embedded directly into ML operations (MLOps). This evolution will empower organizations to innovate responsibly, gain regulator trust, and shorten model development lifecycles. Moreover, vendors will continue to enhance their data governance offerings, providing enterprises with out-of-the-box compliance orchestration for common ML workflows.

ISSN: 3071-5717
PP: 01-11.

9.3. Future Research Directions (e.g., Real-Time Compliance-Aware Feature Stores, AI Explainability with Compliance Alignment)

Several promising directions for future research emerge from this work. One key area is the development of real-time compliance-aware feature stores that dynamically evaluate feature requests against policy rules before serving them to models. These systems would operate like policy firewalls, enforcing access control and lineage verification on-the-fly. Another direction involves tighter integration between AI explainability tools and compliance auditing systems, ensuring that every explanation generated for a model output can be traced to a compliant feature set. Research is also needed on cross-jurisdictional compliance mapping, where data flows and feature engineering must respect multiple overlapping legal regimes. Lastly, there is scope for work in machine-interpretable regulatory policies, enabling end-to-end automation from legal text ingestion to enforcement within ML pipelines. As AI systems become more autonomous, these lines of research will be critical for maintaining lawful and ethical boundaries.

REFERENCES

[1] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Why should I trust you? Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1135–1144). https://doi.org/10.1145/2939672.2939778

[2] Dwork, C., & Roth, A. (2014). The algorithmic foundations of differential privacy. Foundations and Trends in Theoretical Computer Science, 9(3–4), 211–407. https://doi.org/10.1561/0400000042

[3] Google Cloud. (2017). Cloud Data Loss Prevention (DLP) documentation. Google Cloud Platform. Retrieved from https://cloud.google.com/dlp

[4] Microsoft. (2017). Azure Information Protection documentation. Microsoft Learn. Retrieved from https://learn.microsoft.com/

[5] Amazon Web Services. (2017). AWS Identity and Access Management: User Guide. Amazon Web Services. Retrieved from https://docs.aws.amazon.com/iam/

[6] European Union. (2016). General Data Protection Regulation (Regulation (EU) 2016/679). Official Journal of the European Union. Retrieved from https://eur-lex.europa.eu

[7] National Institute of Standards and Technology. (2013). Security and privacy controls for federal information systems and organizations (Special Publication 800-53 Rev. 4). NIST. https://doi.org/10.6028/NIST.SP.800-53r4

[8] PCI Security Standards Council. (2016). Payment card industry data security standard: Requirements and security assessment procedures (Version 3.2). Retrieved from https://www.pcisecuritystandards.org

[9] Basel Committee on Banking Supervision. (2011). Basel III: A global regulatory framework for more resilient banks and banking systems. Bank for International Settlements. Retrieved from https://www.bis.org

[10] Goodman, B., & Flaxman, S. (2017). European Union regulations on algorithmic decision-making and a "right to explanation". AI Magazine, 38(3), 50–57. https://doi.org/10.1609/aimag.v38i3.2741

[11] Pasquale, F. (2015). The black box society: The secret algorithms that control money and information. Harvard University Press.

[12] Veale, M., & Binns, R. (2017). Fairer machine learning in the real world: Mitigating discrimination without collecting sensitive data. Big Data & Society, 4(2), 1–17. https://doi.org/10.1177/2053951717743530

[13] Sweeney, L. (2002). k-Anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 10(5), 557–570. https://doi.org/10.1142/S0218488502001648

[14] Shokri, R., & Shmatikov, V. (2015). Privacy-preserving deep learning. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (pp. 1310–1321). https://doi.org/10.1145/2810103.2813687

[15] Hardt, M., Price, E., & Srebro, N. (2016). Equality of opportunity in supervised learning. Advances in Neural Information Processing Systems, 29, 3315–3323.