Software Engineering Principles and its Applications

Sonakshi Garg

August 28, 2023

My research area is Manifold Learning and Privacy. In today's era, we are loaded with lots of data. Every second abundant amount of data is getting generated through a variety of sources. But still, today's technology faces challenges to process and deal with high amounts of data efficiently. The amount of data produced every day is exponentially increasing. Machine learning algorithms are evolving day by day to provide useful information from this data. With the generation of big data, there also exist enormous high-dimensional data in which the number of instances and attributes is relatively very large, such that data points become very far from each other. This introduces significant challenges in descriptive and exploratory data analysis. The high-dimensional data in today's world exist in many different forms: ranging from tabular data with a higher number of rows and columns, to image data, textual data, etc. When the data has two or three dimensions, graphical plots help in visualizing the local geometry of the data. In contrast, high-dimensional graphs cannot be visualized easily, and plots are less intuitive. Thus, to help the visualization structure of such data, the dimensions of the data must be reduced.

Manifold Learning states that any real-world high-dimensional data set lies on a low-dimensional manifold embedded in a higher-dimensional space [1]. Manifold learning methods are commonly applied in various applications including financial markets and medical images to visualize high-dimensional data. However, the focus of these techniques is on preserving the inherent structure of the data. In recent years, the availability of personal data has become an important concern with respect to privacy-preserving data mining. We are intended in producing valid data mining results without disclosing the underlying private information. Data anonymization is used to minimize disclosure [2]. It reduces or avoids identity disclosure. This aids the data controllers to release and process public data without violating the General Data Protection Regulation (GDPR) policies. Several techniques have been proposed to achieve data anonymization with respect to multidimensional records However, it is still a challenging task, as obtaining highly accurate results requires looking at original values. When the dimensionality of data is high, it becomes even more challenging to preserve the privacy of records while maintaining the local geometry of data.

There are various aspects in software engineering that needs to be considered once any product is built and deployed. One of the important aspect is security and privacy [3]. The security and privacy aspects of software engineering pertain to the considerations, practices, and measures taken during the development, design, and maintenance of software to ensure the protection of data, systems, and users from unauthorized access, attacks, and breaches. Security focuses on safeguarding software and systems against potential vulnerabilities and threats, while privacy concerns are centered around preserving the confidentiality and control of users' personal information and data. Both security and privacy are crucial in building trustworthy and reliable software that meets ethical, legal, and regulatory standards, and they play a vital role in maintaining user trust and the integrity of software systems in today's interconnected digital landscape.

In this new era of technology, a more focus is shown in decentralized learning framework, where the data is distributed between different participants and each participant trains its data on a local model without sharing with each other. It offers a lot of advantages, privacy being the most important one. Federated Learning allows training machine learning models without the need to share raw data centrally. This ensures that sensitive data stay with the participants, which leads to reduced risk of data breaches. This is one of the application of software engineering where security and privacy is very strongly considered [4].

This application is also related with my research field. In my research we also aim to protect the personal information from adversary using any privacy models. When the personal data that needs to be protected is high-dimensional, then in order to not loose information, my research focuses on using manifold learning which helps to preserve the inherent structure of data. It also help to deal with existing privacy models more easily using manifold learning technique.

Another important aspect in software engineering principles is quality assurance [5].It refers to the systematic process of ensuring that software products and processes meet the required standards and expectations. It involves various activities aimed at preventing defects, identifying issues, and improving the overall quality of software. Some of these activities like requirements analysis, test planning, design, and execution, as well as defect management, regression, performance, and security testing. By integrating these practices throughout the software development life cycle, organizations can enhance reliability, performance, and user satisfaction while reducing the risk of defects and costly rework.

Any deliverable or product can not be deployed unless it undergoes from quality assurance. It plays a crucial role in detecting the defects in code. This helps to rectify the defects early in the development phase reducing the likelihood of issues in final product. ALso, it leads to cost savings when the defects are identified and fixed early, a lot of maintenance cost can be saved which could have occured otherwise. Quality assurance also mitigates risks associated with software failures, security breaches and loss of data, safeguarding both the user and the organization. Another important principle of software engineering is Regulations and compliance. In software engineering, it refers to the set of rules, standards, guidelines, and legal requirements that software developers and organizations must follow when developing and deploying software applications [6, 7]. These regulations are often put in place to ensure the safety, security, privacy, and ethical use of software systems. Few aspects of regulations and compliance in software engineering are Data Privacy and Protection: Regulations such as the General Data Protection Regulation (GDPR) in the European Union and the California Consumer Privacy Act (CCPA) in the United States impose strict requirements on how personal data is collected, stored, processed, and shared. Software applications that handle personal data must adhere to these regulations to protect user privacy. Security Standards: Various industries have specific security standards that software systems must meet. For example, the Payment Card Industry Data Security Standard (PCI DSS) sets requirements for handling payment card information, and the Health Insurance Portability and Accountability Act (HIPAA) establishes standards for protecting healthcare data. Accessibility: Many countries have laws and standards that mandate accessible software and websites for people with disabilities. For instance, the Web Content Accessibility Guidelines provide guidelines for making web content accessible to a wide range of users. To navigate these regulations and ensure compliance, software engineering principles often need to incorporate compliance practices into their development processes. This can involve thorough testing, documentation, security assessments, and working closely with legal and regulatory experts to ensure that software systems meet the necessary standards and requirements.

With the growing use of artificial intelligence and machine learning in software, there was a focus on understanding how AI systems can comply with regulations and ethical standards. Research was exploring methods to ensure that AI-driven decisions are transparent, explainable, and compliant with laws like GDPR. Also, Privacy regulations were pushing for the development of techniques that allow software to function while minimizing the exposure of personal data. Research was ongoing in the areas of differential privacy, data anonymization, and secure multi-party computation. Beyond mere compliance, the ethical considerations of software development were receiving more attention. Researchers were exploring ways to integrate ethical considerations into the software development process to avoid biases, discrimination, and other ethical pitfalls.

References

[1] Vin Silva and Joshua Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. *Advances in neural information processing systems*, 15, 2002.

[2] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International journal of uncertainty, fuzziness and knowledge-based systems*, 10(05):557–570, 2002.

[3] Lin Liu, Eric Yu, and John Mylopoulos. Security and privacy requirements analysis within a social setting. In *Proceedings. 11th IEEE International Requirements Engineering Conference, 2003.*, pages 151–161. IEEE, 2003.

[4] Yifei Zhang, Dun Zeng, Jinglong Luo, Zenglin Xu, and Irwin King. A survey of trustworthy federated learning with perspectives on security, robustness, and privacy. *arXiv preprint arXiv:2302.10637*, 2023.

[5] Samuel Daniel Conte, Hubert E Dunsmore, and YE Shen. *Software engineering metrics and models.* Benjamin-Cummings Publishing Co., Inc., 1986.

[6] Yod-Samuel Martin and Antonio Kung. Methods and tools for gdpr compliance through privacy and data protection engineering. In *2018 IEEE European symposium on security and privacy workshops (EuroS&PW)*, pages 108–111. IEEE, 2018.

[7] Shareeful Islam, Haralambos Mouratidis, and Jan Jürjens. A framework to support alignment of secure software engineering with legal regulations. *Software & Systems Modeling*, 10(3):369–394, 2011.