Software Engineering Assignment

By: Ayush Kumar Varshney

1 My Ongoing Research

In today's world, Artificial Intelligence (AI) plays a crucial role in our day-to-day life. AI techniques are widely used in object recognition, speech recognition, medical imaging, robotics and many other fields. AI approaches and Machine Learning (ML) in particular are very data hungry. They tend to improve with the quality and quantity of data. The data often includes sensitive and personal information which must be guarded to ensure security/privacy of each individual or organization. Several guidelines exist such as Europe's General Data Protection Regulation (GDPR), to regulate the use of data in ML. My research project focuses on developing privacy-preserving statistical and ML algorithms which avoids various types of disclosure risk associated with the leak of personal information for an individual or a group of individual. Data anonymization is one such way of avoiding disclosure risk. It ensures that personal identifiable information is permanently changed so that the data subject can no longer be directly or indirectly identified, either by the data controller alone or in partnership with any other party. This protects a person's identity. For the data to meet the criteria for data anonymization, it must be difficult or impossible to trace it to a specific person.

There exists several data masking and privacy-preserving models such as k-anonymity, differential privacy, integral privacy, etc. which tries to protect privacy of individual or organization from any adversaries. Adversaries aim to gain sensitive information about individual or a group of individuals inferring from ML models. My current work focuses on generating integral privacy model which can be generated by multiple disjoint datasets. The basic idea is datasets which have similar data distribution leads to similar model parameters. This allows us to choose a model which has can be generated by multiple clients. In our project, we also explore the concept drift detection in online learning. Concept drift implies that the statistical properties of the data may change over time, necessitating the model to adapt to these changes to ensure reliable predictions. In the literature of concept drift, almost none of the approaches focus on the privacy perspective of drift detection. In our approach, we have considered Integral Privacy as an alternative to DP to generate high utility, privacy-preserving machine learning models. Integrally private models provide sound defense against various attacks. We have worked on proposing a concept drift detection methodology using ensemble of integrally private models in our

latest work.

2 Discussions from software engineering lectures

Automated testing involves using software tools instead of the manual work of validating software products. It aims to enhance the effectiveness, execution speed, and coverage of software testing by reducing the number of test cases that need to be run manually, it can also help in reducing the cost of validating the steps in software engineering. Subsequent development cycles necessitate the repeated execution of the same test suite, which can be recorded and replayed as needed using test automation tools. These tools may also generate comprehensive test reports and compare expected and actual results. Automation in software testing is crucial for several reasons: (1) manual testing is time-consuming, costly, and prone to errors; (2) automation does not require human intervention during software testing; and (3) it increases test coverage and the speed of test execution. Automated testing allows for a leaner quality assurance (QA) team and enables the team to concentrate on more critical features. Tests such as end-to-end, unit, integration, and performance tests should be automated first. However, automated testing is not suitable for scenarios where requirements frequently change or test cases are executed on an ad-hoc basis. In such cases, manual testing is necessary. Machine learning (ML) can enhance the procedures of automated software testing. Various ML applications such as object detection , anomaly detection, and pattern recognition can be integrated into various steps of software design in order to get higher utility. For instance, object detection can be utilized in the automation of user interface testing, where the human eye may miss certain defects on a page. For API monitoring, anomaly detection can be instrumental in identifying unusual API events and traffic. These cases makes stronger case for both SE4ML and ML4SE. At the same time, these automated steps can not be trusted blindly because overall in the present stage of machine learning it is a very black-box methodology and hence human verification or supervision is a must.

Data and security breach is a substantial issue in the software engineering pipeline. More and more countries are implementing data regulatory bills such as GDPR. Even a small security or data breach can cost a company a significant amount of money. From machine learning point of view, significant amount of data (sometimes private and sensitive as well) is used to train the machine learning models. Security breaches such as unathorized access to data, trained model or even gradients during training can cause a huge data leakage. At the same time with access to models and data, attackers can introduce seemingly not harmful malware to produce false predictions. As a result, it is essential to maintain model integrity and data security throughout the software engineering pipeline for the effective and secure deployment of machine learning systems. For this reason, data privacy models such as k-anonymity, differential privacy becomes a must for machine learning models.

DevBots in AI4SE refers to the integration of AI methods such as machine

learning, natural language processing in order to automate various tasks in the software development process. DevBots can generate complete or snippets of codes based on the requirements and design such as Github co-pilot which can generates code snippets based on the comments. They are also very helpful in reviewing the code and requirements, they can highlight any inconsistency or violations and can even suggest optimization. As mentioned for automated testing, we use DevBots for automated testing as well. However, it is important to note that while DevBots can greatly assist in the software development process, they are not a replacement for human developers.

Sentiment analysis in software engineering use natural language processing in order to analyze and improve the sentiments in various artifacts such as issue reports, comments in the code and reviews. Consider an example for quality assurance, youtube application in Appstore has close to 4 million reviews, manual assessment of sentiments would be very labour intensive, it will cost time and money. On the other hand, with the help of NLP it becomes very easy to assess the comments in order to improve the quality in the next iterations of the applications. However, from privacy's point of view, reviews of any artifact which can have any personal information must be preprocessed and NLP models must not be trained on such data.

There has been a significant increase in the efforts for compliance to ensure the security, integrity and legality of software applications. Security involves measures to avoid unauthorized access, data breaches and cyberattacks. This calls for frequent and additional malware checks during the data collection, data verification, training the ML systems and monitoring phases. Compliance on the other hand, requires that the software complies with accepted rules, regulations, and standards, which might differ greatly based on the location, sector, or type of application. Specially in the age of big data, where multiple parties want to leverage data but struggle to share data due to compliance. These areas presents an interesting future direction.

3 Future trends and directions in software engineering

My PhD work investigate the privacy-preserving models. It is generally assumed that privacy comes at the cost of utility/accuracy. The literature of the privacy-preserving models focuses on either anonymizing the data or providing privacy through noise addition. We are looking to find more generalized models, which doe not cost much utility. And with the current wave of AI, more and more AI algorithms will find their applications in the software engineering. But AI algorithms specially machine learning algorithms have big privacy and security concerns. So, our work aim to contribute in proposing improved privacy-preserserving solutions.