

## Security Code Neural Network

### Hintergrund

Um das Wissen über existierende Schwachstellen der Allgemeinheit bekannt zu machen, existieren Datenbanken denen dieses Wissen innewohnt. Für einen Großteil dieser Schwachstellen gibt es Quellcode Beispiele welche diese Schwachstellen enthalten. Wird während der Entwicklung ähnlicher Quellcode verwendet, so wird ebenfalls eine Schwachstelle ins Projekt eingefügt. Für einen Menschen ist es nahezu unmöglich alle Schwachstellen inkl. deren Quellcodes zu kennen und während der Entwicklung zu berücksichtigen.

Ein Verfahren um Wissen, welches in großen Datenmengen enthalten ist zu nutzen, ist ein Ansatz der künstlichen Intelligenz in Form von Neuronalen Netzen. Es existieren unterschiedliche Herangehensweisen diese zu nutzen. Dies reicht von einfachen einzelnen neuronalen Netzen bis hin zur Kombination derer. Es gibt bereits zahlreiche Frameworks, wie das Open Source Framework TensorFlow, die bei der Implementierung von Ihnen in der Programmiersprache Java unterstützen [1].

[1] <https://www.tensorflow.org/>

### Aufgabe

Im Rahmen dieser Arbeit soll ein Eclipse Plugin entwickelt werden, dass Ansätze neuronaler Netze nutzt um Programmcode zu erkennen, welcher über eine Schwachstelle verfügt. Dem Softwareentwickler soll während der Entwicklung ein Hinweis erscheinen, sobald er Quellcode nutzt, welcher vom neuronalen Netz als ähnlich zu gefährdeten Programmcode erkannt wird. Dies soll Ihnen eine zusätzliche Unterstützung zur Programmcode Schwachstellenanalyse liefern. Hierfür soll das Wissen von bestehenden (Code-) Datenbanken genutzt werden. Bestehende anwendungsformen von neuronalen Netzen müssen auf Basis der Effizienz analysiert und verglichen werden um bestmögliche Ergebnisse zu erzielen (Recall, Precision etc.). Weiterhin muss sich ein Konzept der Darstellung überlegt werden, welches diese Befunde dem Entwickler mitteilt. Die Analyse des Programmcodes soll fortlaufend während der Programmierung durchgeführt werden und darf die Arbeit des Entwicklers nicht negativ beeinflussen (Keine Ladezeiten/Stopper) während der Entwicklungsphase. Weiterhin sollten die Problembereiche im Editor der Entwicklungsumgebung hervorgehoben werden.

#### Anforderungen:

- Literatursuche nach Ansätzen zu neuronalen Netzen in Bezug auf Source Code
- Analyse bestehender Ansätze von neuronalen Netzen
- Gut strukturierter und kommentierter Programmcode (Modular gekapselt)
- Entwicklung eines Eclipse Plugin
- Analyse des Recall & Precision Verhältnisses unterschiedlicher neuronaler Netze
- Evaluation des erstellten Ansatzes (Precision & Recall)
- Hoher Java Programmieraufwand

### Organisatorisches

**Betreuer:** M. Sc. Fabien P. Viertel, [fabien.viertel@inf.uni-hannover.de](mailto:fabien.viertel@inf.uni-hannover.de), Raum G307

**Prüfer:** Prof. Dr. Schneider **Beginn:** Ab sofort möglich