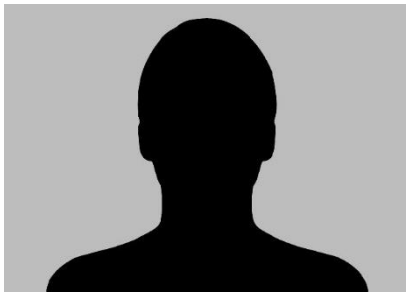
Zürcher Hochschule
für Angewandte Wissenschaften

School of Engineering

InIT Institut für angewandte Informationstechnologie

Machine Learning: Finden von Schwachstellen in Quellcode

In dieser Arbeit wird die Vorhersage von Schwachstellen in C++ Quellcode von Mozilla untersucht. Das Hauptziel ist die Replikation von "Vulture", einem Tool das von Neuhaus et al. im Rahmen von "Predicting Vulnerable Software Components" entwickelt wurde. Das Tool sammelt Daten aus Bugtrackern und Repositories, um Komponenten der Software bestehende Schwachstellen zuzuweisen. Mit diesen Informationen und den Features der Komponenten, wie z.B. Function Calls, trainiert es eine Support Vector Machine, um Vorhersagen über weitere Schwachstellen zu treffen.
Da die Daten von Vulture nicht mehr existieren, wurde der Ansatz so ähnlich wie möglich implementiert und die Ergebnisse mit Vulture verglichen. Obwohl mehrere grosse Softwareprojekte geprüft wurden, fiel der Entscheid ebenfalls auf Mozilla. Es wurden vergleichbare Resultate wie bei Vulture erreicht, weshalb die Replikation als erfolgreich eingestuft wird. Es wurde ebenfalls mit zusätzlichen Informationen experimentiert. Die vergangenen Features der Komponenten wurden extrahiert, bevor deren Verwundbarkeiten behoben wurden. Alle so extrahierten Features wurden zu den Daten hinzugefügt, sofern dort noch kein identischer Eintrag existierte. Das Experiment resultierte in deutlich höheren Recall-Werten bei allen Arten von Features. Dennoch lässt sich daraus schliessen, dass die Werte unzuverlässig sind und sich nicht als Indikatoren für die echte Qualität von solchen Modellen eignen. Weiter wurden zusätzliche Features wie Namespaces und Preprocessing Macros evaluiert. Bei allen getesteten Features wurde eine gute Precision ermittelt. Bei den Recall-Werten liefern allerdings nur Includes und Function Calls, wie sie auch bei Vulture verwendet wurden, brauchbare Ergebnisse. Die Vorhersagen der Modelle wurden ebenfalls mit realen Daten evaluiert. Dazu wurden Modelle mit vergangen Zuständen des Repositories trainiert. Anschliessend wurden deren Vorhersagen mit Daten von späteren Zeitpunkten validiert. Die Ergebnisse dieses Experiments waren ernüchternd, dennoch waren die Vorhersagen deutlich besser als eine zufällige Auswahl.
Zum Schluss wurde Support Vector Machines und Decision Trees miteinander verglichen. Die Precision ist bei SVMs deutlich höher und das Trainieren des Modells benötigt weniger Zeit. Beim Recall-Wert und bei der Laufzeit für Vorhersagen sind die beiden vergleichbar. Ausserdem sind Decision Trees analysierbar, was genutzt werden kann, um das wichtigste Feature zu ermitteln.



Diplomierende
Hannes Klauser
Aurelio Malacarne

Dozierende
Mark Cieliebak
Stephan Neuhaus

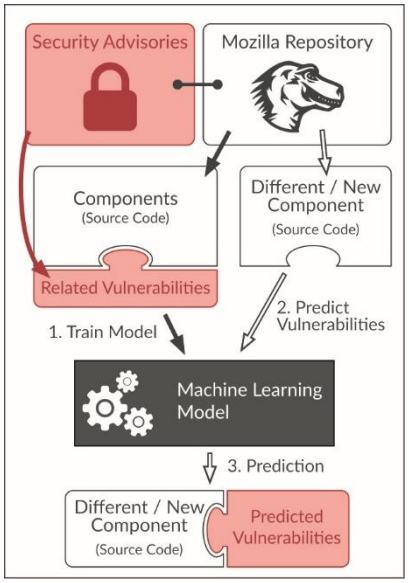


Illustration des gesamten Ablaufs. Informationen aus dem Quelltext und aus Security-Advisories werden zu Komponenten verknüpft, um ein Machine-Learning Modell zu trainieren. Dieses Modell soll Schwachstellen in weiteren Komponenten voraussagen.