

Erstellung einer Webseite und Smartphone-App zur Steuerung einer internetfähigen Kamera (Schwenken, Zoomen der Kamera,...).

Aufbau eines kleinen Netzwerkes zur **Untersuchung des Protokolls MQTT** – ein wichtiges Protokoll im Internet of Things.

Erstellung einer grafischen Animation (mit Qt in C++) zur Darstellung der Paketweiterleitung durch einen Router mit Warteschlangen, zum Vergleich leitungsvermittelter Telefonnetz / paketvermittelter Internet oder zur Demonstration des Unterschieds statisches / statistisches Multiplexing.

Demonstration und Programmierung **verschiedener Verschlüsselungsverfahren** in C oder C++ unter Verwendung von Bibliotheken für die Verschlüsselungsverfahren selbst.

Einführung von IPv6 im Rechnernetz des ‚IT-Lab‘ mit Erarbeitung von typischen IPv6 Demobeispielen (z.B. Privacy Extension, Mobile IP, DHCPv6).

Programmierung eines **NAO-Roboters** (humanoider Roboter) in C oder C++

Untersuchung netzwerktypischer Parameter des Netzes im 'IT Lab' mit einem **Netzwerkanalysator**.

Untersuchung des SNMP Protokolls für das **Netzwerkmanagement** im IT-Lab u.a. mit Wireshark.

Installation von WLAN-fähigen Telefonen und Analyse des Aufbaus einer WLAN-Verbindung und eines Cell-Handovers mit dem Protokollanalysator Wireshark.

Gerne weitere Themen für Abschlussarbeiten im Bereich der Softwareentwicklung nach Absprache.

Didaktische Konzepte zum Erlernen der Programmiersprache Python im Zusammenhang mit der Vektorisierung numerischer Algorithmen für Studierende mit Vorkenntnissen in C/C++

Python hat sich als Programmiersprache im Kontext von Data Science und maschinellem Lernen durchgesetzt und bietet Anwendern einen benutzerfreundlichen Zugang zu zahlreichen Frameworks für Datenanalysen, Datenvisualisierungen und zu effizient implementierten Algorithmen für numerische Berechnungen.

Um diese effizient implementierten Algorithmen in Python nutzen zu können, müssen häufig iterative Algorithmen für mehrdimensionale Array-Operationen umgeschrieben werden. Das erfordert ein gewisses Umdenken im Umgang mit der Programmiersprache und den zugrundeliegenden Algorithmen, vor allem wenn man bisher ausschließlich mit C/C++ gearbeitet hat und in diesem Zusammenhang nicht bereits die Bibliotheken BLAS und LAPACK verwendet hat.

Studierende mit Vorkenntnissen in C/C++, die sich gerne im Bereich des maschinellen Lernens weiterbilden möchten, erwarten also gleichzeitig mehrere Herausforderungen. Zum einen müssen sie eine neue Programmiersprache, mit der damit verbundenen neuen Syntax, lernen und sich zum anderen gleichzeitig mit dem Konzept der Vektorisierung numerischer Algorithmen vertraut machen.

Ziel dieser Abschlussarbeit soll es sein, didaktische Konzepte und Lernmaterialien zu entwickeln, die Studierenden mit den beschriebenen Vorkenntnissen den Einstieg in das Thema "Maschinelles Lernen mit Python" erleichtern. Dabei stehen folgende Themenschwerpunkte im Vordergrund:

- Umgang mit dem Python-Ökosystem, Paketmanagern und virtuellen Umgebungen
- Vektorisierung numerischer Algorithmen mit NumPy
- objektorientierte Programmierung mit Python (optional: bei Vorkenntnissen in C++ können gerne die Unterschiede zu objektorientierter Programmierung mit C++ betrachtet werden)
- Hauptunterschiede zwischen Python und C/C++
- Auf welche Sprachkonstrukte der Programmiersprache Python sollte man sich im Hinblick auf einen schnellen und effizienten Einstieg in den Bereich des maschinellen Lernens vorrangig konzentrieren?
- Umgang mit Frameworks wie scikit-learn, TensorFlow und PyTorch

Dabei können auch gerne externe digitale Lernplattformen (wie z.B. codecademy.com oder codewars.com) analysiert und die didaktische Qualität dieser Plattformen im Hinblick auf die beschriebene Fragestellung verglichen werden. Programmieren lernt man am besten durch Übung. Bieten solche Plattformen Übungsmaterial an, welches für unsere Fragestellung geeignet ist?