Hochschule Reutlingen

Reutlingen University

- Studiengang Mechatronik Bachelor -

Bachelor–Thesis

Künstliche Intelligenz und Datensicherheit im Internet der Dinge

Dennis Fuchs
Heppstr. 42
72760 Reutlingen

Matrikelnummer: 741468

Betreuer:	Prof. Dr. rer. nat. Stefan Mack
Zweitbetreuer:	Prof. Dr. rer. nat. Eberhard Binder
Abgabedatum:	02.02.2018

Danksagung

An dieser Stelle möchte ich mich bei all denjenigen bedanken, die mich während der Anfertigung dieser Bachelor-Thesis unterstützt und motiviert haben.

Ganz besonders möchte ich Herrn Prof. Dr. rer. nat. Stefan Mack danken, der meine Arbeit durch seine fachliche und persönliche Unterstützung begleitet hat.

Darüber hinaus möchte ich mich bei meinen Betreuern Anian Bühler und Michael Hermann bedanken, durch deren Anregungen und Ideen meine Arbeit kontinuierlich verbessert wurde.

Ein besonderer Dank gilt meinen Eltern, die mir durch ihre Unterstützung mein Studium ermöglicht haben.

Abstract

Through the internet of things more and more devices get connected. Nearly all of them simplify the everyday life, but they produce a lot of sensitive personal data. This data is useful for analyses through artificial intelligences and interesting for cybercriminals. Thereby, cybercrime is rising constantly. Furthermore, it's not possible to shop online without getting analyzed by an artificial intelligence.

On the one hand, pupils who participate in the project letgoING should be able to experience the advantages of the internet of things with a newly developed setup. On the other hand, pupils should also think about security while surfing the internet and know common methods of cybercriminals to crack or attack systems. The thesis deals with analyzing the newly setup and attacking the IP-communication. The attacks will be tested on the given setup.

Moreover, pupils should be able to understand the basic concepts of artificial intelligence. Therefore, this thesis develops different ideas about how artificial intelligence could be integrated in the current letsgoING content. After choosing the most practicable project, the necessary software will be developed and the generated artificial intelligence will be tested. To conclude the topic the gain of using an artificial intelligence instead of the conventional technical solution will be worked out.

Inhaltsverzeichnis

Kapitel 1

Einleitung

Im Zuge der immer schneller fortschreitenden Digitalisierung werden zunehmend mehr Geräte mit dem Internet vernetzt. Dadurch entstehen viele Erleichterungen im geschäftlichen, aber auch im privaten Umfeld. Durch Anwendungen wie SmartHome kann die eigene Wohnung rund um die Uhr und von überall vollautomatisiert gesteuert oder überwacht werden. Ein weiteres Beispiel sind zahlreiche Sprachassistenten die Aufgaben, wie den nächsten Supermarkt finden, Musik abspielen oder das aktuelle Kinoprogramm heraussuchen, auf Sprachkommando erledigen. Durch diese komfortablen Assistenzsysteme rückt der Alltag immer mehr in die digitale Sphäre des Internets. Selbst die Erledigung alltäglicher Aufgaben wie das Arbeiten kann inzwischen über das Internet erledigt werden. Gleichzeitig steigt die Menge an privaten, sensiblen und firmeninternen Daten, die über das Internet kommuniziert werden. Außerdem sind die Benutzer für die Erleichterungen häufig bereit, Programmen und Anwendungen immer mehr Berechtigungen zu erteilen. Es entstehen digitale Fußabdrücke und riesige Datenmengen.

Diese Datenmengen bieten einen optimalen Anwendungsfall für künstliche Intelligenzen. Durch gezielte Datenanalysen können Nutzerverhalten studiert und Angebote auf die Nutzer angepasst werden. Viele Internetseiten nutzen inzwischen Tracking- und Analysesysteme, um ihre Artikel, Werbungen und Marktanalysen auf die Endkunden anzupassen.

Außerdem bietet die Kommunikation sensibler Daten über das Internet und die steigende Vernetzung in Kombination mit einem unzureichenden Bewusstsein für Datensicherheit idealen Nährboden für Cyberkriminalität. Die Anzahl der Straftaten im digitalen Raum nehmen stetig zu und immer neue Programme überlisten die Sicherheitsmechanismen von Computern. Ein derzeit unzureichendes Bewertungs- bzw. Zertifizierungssystem für IT-Sicherheit von Produkten unterstützt die steigende Anzahl von Straftaten im Netz.

Im Rahmen dieser Arbeit sollen beide Themenbereiche für das Kooperationsprojekt letsgoING aufbereitet werden. Das Projekt soll bei Schülern die Begeisterung für Technik und ingenieurwissenschaftliche Studiengänge wecken. Da in den Schulen immer aktuelle Themen behandelt werden sollen, müssen die vorhanden Module weiterentwickelt werden.

Zum Thema künstliche Intelligenz soll ein Modul entwickelt werden, welches den Schülern erste Einblicke in die Funktionsweise künstlicher Intelligenzen ermöglicht. Das Modul soll sich mit einem Teilgebiet der künstlichen Intelligenz, den sogenannten neuronalen Netzen, beschäftigen. Verschiedene Ansätze für ein Modul sollen verglichen und die praktikabelste Variante realisiert werden. Abschließend soll die Funktionsweise im Vergleich zu herkömmlichen Lösungsmethoden analysiert werden.

Der Bereich Datensicherheit beschäftigt sich mit Angriffsmöglichkeiten auf ein neu entwickeltes Modul aus dem Themengebiet Internet der Dinge. In diesem Modul tauschen mehrere WLAN-Module mit einem Server über eine IP-Kommunikation Daten aus. Die Sicherheit des Systems soll erforscht und Schwachstellen aufgedeckt werden. Aus den Schwachstellen werden Angriffsmöglichkeiten erarbeitet, die auf ihre Funktionalität getestet werden und im Unterricht vorgeführt werden können. Ziel ist es, die Schüler für Schwachstellen und Sicherheitsaspekte des Internets zu sensibilisieren. Durch diese Sensibilisierung soll eine digitale Mündigkeit gefördert werden.

Kapitel 2

Grundlagen

In diesem Kapitel werden verschiedene Grundlagen für die weitere Ausarbeitung erläutert. Diese erstrecken sich von der Kommunikation im Internet, über ausgewählte Themen der Datensicherheit bis hin zu verschiedenen Aspekten künstlicher Intelligenz.

2.1 Einführung in die IP-Kommunikation

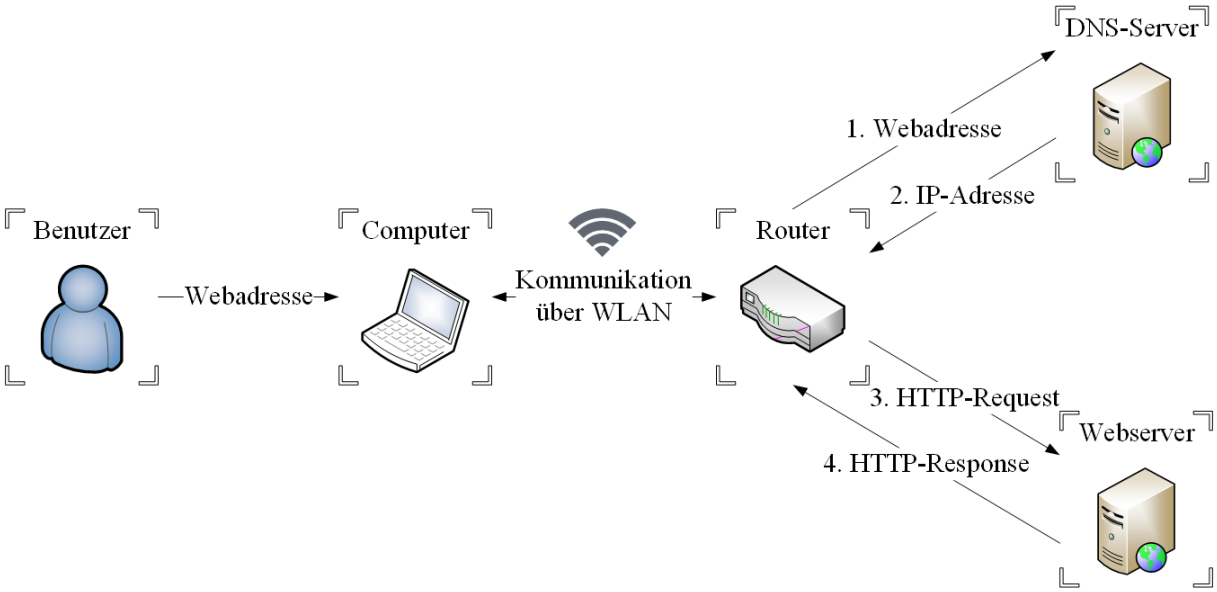

Abbildung 2-1 - Aufruf einer Webadresse im Internet

Die Entwicklung des Internets lässt sich auf die Entstehung des ersten Rechnernetzes im Jahr 1969 zurückverfolgen. Das sogenannte ARPANET wurde für die amerikanische Regierung als Kommunikationssystem für den Kriegsfall erforscht. Verschiedene Rechner in den USA konnten auch nach Ausfall einzelner Rechner und Netze weiterhin miteinander kommunizieren. Diese Robustheit sollte die Kommunikation im Kriegsfall sichern. Außerdem wurde dieses Rechnernetz von Universitäten als Forschungs- und Wissenschaftsnetz genutzt. Bereits in der Entwicklungsphase basierte der Informationsaustausch auf einzelnen Datenpaketen. Neben dem amerikanischen ARPANET gab es auch bereits in Europa ein aus einzelnen nationalen Netzen bestehendes europäisches Netzwerk. Durch die steigende Internationalisierung der

Forschergruppen wurde die Notwendigkeit eines internationalen Netzwerkes zum Austausch von Ergebnissen verstärkt.

Auf Grundlage der Idee von Tim Berners-Lee wurde das weltweilt verteilte System unter dem Namen World Wide Web erstellt. Dieses System verbindet die nationalen und kontinentalen Netze zu einem Geflecht von Rechnern, die als Webserver fungieren. Jeder Web-Inhalt ist unter einer einheitlichen Web-Adresse, der sogenannten URL, erreichbar. Durch Eingabe einer URL in einen Web-Browser kann auf den zugehörigen Web-Server mit seinen Web-Inhalten zugegriffen werden. Hypertext Markup Language HTML wurde als abstrakte Sprache für die Darstellung von Webseiten eingeführt. Das Protokoll Hypertext Transfer Protocol HTTP wurde für den Transport von Web-Inhalten zwischen Browsern und Webservern etabliert. Der Browser sendet eine Anforderung, einen sogenannten HTTP-Request, an den durch die URL spezifizierten Webserver. Dieser antwortet daraufhin mit einer HTTP-Response. In der Antwort befindet sich der angeforderte Web-Inhalt. Für die Übertragung wurden noch die Protokolle Transmisson Control Protocol TCP und User Datagram Protocol UDP festgelegt. Die beiden Protokolle bauen auf dem Internet Protocol IP auf. Diese Normierungen bilden die Grundlage für das heutzutage bekannte Internet. [1]

In der Abbildung 2-1 ist diese Kommunikation dargestellt. Der Benutzer gibt eine Web-Adresse in den Webbrowser seines Computers ein. Der Computer ist über WLAN mit dem Router, welcher an das Internet angeschlossen ist, verbunden. Zuerst wird die Web-Adresse an einen sogenannten DNS-Server gesendet. Dieser übersetzt die Web-Adresse in eine IP-Adresse und übermittelt diese zurück an den Computer. Anschließend benutzt der Computer die IP-Adresse, um einen HTTP-Request an den Webserver zu stellen. Der Webserver verarbeitet den HTTP-Request und überträgt den Inhalt der angeforderten Webadresse durch einen HTTP-Response zurück an den Computer. Der Benutzer sieht nun auf seinem Browser den Webinhalt, der unter der verwendeten Webadresse verfügbar ist.

2.1.1 Aufbau von Computernetzwerken

Ein Netzwerk ist die Verbindung von beliebig vielen Rechnersystemen. Das kleinste mögliche Netzwerk sind zwei Computer, die durch ein Übertragungsmedium wie beispielsweise WLAN oder ein Ethernetkabel, verbunden sind. Das größte derzeit verwendete Netzwerk ist das World Wide Web. In einem Netzwerk wird jedem Teilnehmer eine eindeutige Netzwerkadresse zugeordnet, damit dieser für die anderen Teilnehmer erreichbar ist. Netzwerkadresse in Netzwerken, die auf dem Internet Protocol basieren, werden als IP-Adresse bezeichnet. Zum

Beispiel in einem WLAN-Netzwerk wird die IP-Adresse durch den Router vergeben. Der Router übernimmt die Aufgabe, dass jeder Teilnehmer eine eindeutige IP-Adresse. [2]

Die IP-Adresse besteht aus vier zusammenhängenden Zahlen zwischen 0 und 255. Ähnlich wie bei einer Postanschrift werden die Zahlen einer IP-Adresse immer detaillierter. Die Postanschrift wird von dem jeweiligen Land über das Bundesland und die Stadt bis hin zur Straße immer weiter verfeinert. Auch die IP-Adresse ist nach diesem Prinzip aufgebaut. Bei der IP-Adresse 192.168.2.10 bezeichnet die Zahl an vierter Stelle die Adresse des Computers. Die ersten drei Zahlen geben übergeordnete Netzwerke, wie bei der Postadresse die Stadt, an. [3]

Das Internet ist ein weltweiter Verbund von Rechnernetzwerken, demnach handelt es sich um ein globales Netzwerk. Wie bereits erläutert gibt es aber auch kleinere Netzwerke, die nicht mit dem Internet verbunden sind. Diese werden als lokale Netzwerke bezeichnet und entstehen durch die Verbindung verschiedener Rechnersysteme mit geeigneten Übertragungsmedium. Sowohl globale als auch lokale Netzwerke verwenden die gleichen Protokolle und Übertragungsmedien. Daher können lokale Netzwerk auch an das globale Netzwerk angeschlossen werden. Beispielsweise ist ein WLAN ein lokales Netzwerk, welches durch den Router geöffnet und verwaltet wird. Dieses wird durch den Anschluss des Internetanbieters mit dem globalen Netzwerk verbunden. Ohne diese Verbindung könnten die einzelnen Rechner im lokalen Netzwerk miteinander kommunizieren, aber nicht auf Rechner außerhalb dieses Netzwerkes zugreifen. [4]

2.1.2 Address Resolution Protocol

Das Address Resolution Protocol ARP übersetzt die IP-Adresse in die MAC-Adresse. Bei der MAC-Adresse handelt es sich um eine einzigartige Hardwareadresse jedes Netzwerkadapters. Sämtliche lokale Netzwerke nutzen die Hardwareadresse anstatt der IP-Adresse, um die Daten zu adressieren. Vor dem Versenden von Daten wird eine Nachricht mit der IP-Adresse des Empfängers und der MAC-Adresse ff:ff:ff:ff:ff:ff an alle Netzwerkteilnehmer verschickt. Eine Nachricht, die an alle Netzwerkteilnehmer gerichtet ist, wird auch Broadcast genannt. Der Teilnehmer des Netzwerks, dem diese IP-Adresse zu geordnet ist, antwortet mit seiner Hardwareadresse. Diese Adresse wird temporär in einer lokalen ARP-Cache gespeichert. Dadurch muss die ARP-Adressauflösung nicht vor jedem Versenden durchgeführt werden und die Kommunikation kann somit beschleunigt werden. [5]

Die ARP-Cache wird nur temporär angelegt, damit Netzwerke dynamisch auf Änderungen reagieren können. Zum Beispiel könnte ein Client die Verbindung zum Netzwerk verlieren und bei der erneuten Anmeldung eine andere IP-Adresse erhalten. Da die IP- und MAC-Adressen

nicht statisch zugeordnet sind, stellt diese Änderung der IP-Adresse kein Problem für die Kommunikation dar. Die ARP-Cache wird lediglich aktualisiert. [6]

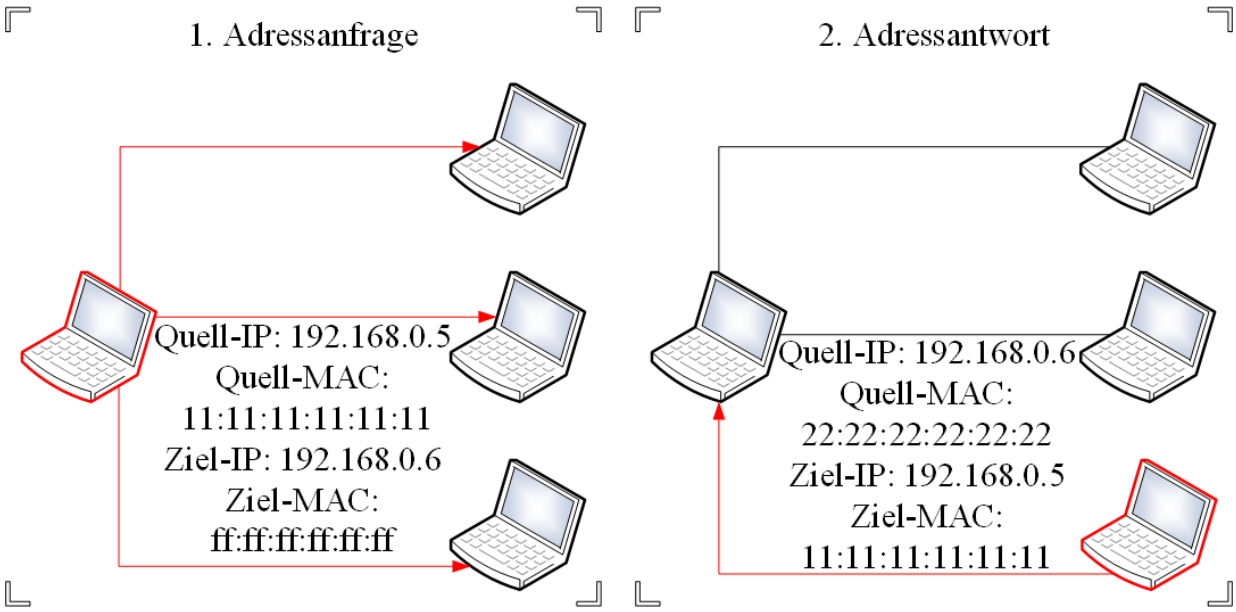

Abbildung 2-2 - ARP-Protokoll

In der Abbildung 2-2 wird dieser Prozess beschrieben. Der auf der linken Seite rot markierte Rechner will von der IP-Adresse 192.168.0.6 die MAC-Adresse erfahren. Dieser Broadcast wird im ARP als Adressanfrage oder ARP-Request bezeichnet. Der angesprochene Rechner antwortet auf der rechten Seite der Abbildung mit seiner MAC-Adresse 22:22:22:22:22:22. Die Antwort wird als Adressantwort oder auch ARP-Reply bezeichnet.

```
C:\Users\Dennis>arp -a

Schnittstelle: 192.168.2.119 --- 0x10
  Internetadresse      Physische Adresse     Typ
  192.168.2.1          d4-21-22-64-7e-a0     dynamisch
  192.168.2.255        ff-ff-ff-ff-ff-ff     statisch
```

Abbildung 2-3 - ARP-Cache eines Rechners mit Windows Betriebssystem

Die Abbildung 2-3 ist die ARP-Cache eines Rechners mit dem Betriebssystem `Windows` dargestellt. Mit dem Befehl „`arp -a`“ kann diese unter `Windows` ausgelesen werden. Die Schnittstelle bezeichnet die IP-Adresse des Rechners. Dieser kennt unter der IP-Adresse 192.168.2.1 den Router des Netzwerks und ordnet ihm die physische bzw. MAC-Adresse d4:21:22:64:7e:a0 dynamisch zu. Die zweite Adresse bezeichnet die größte mögliche IP-Adresse in einem Netzwerk und wird der MAC-Adresse ff:ff:ff:ff:ff:ff zugeordnet. Diese IP-Adresse wird als Broadcastadresse in IP-Netzwerken reserviert und wird daher statisch in der ARP-Cache zugeordnet. [4]

2.1.3 Internet Protocol, Transmission Control Protocol und User Datagram Protocol

Besonders wichtig für das Internet ist das Internet Protocol IP. Die wichtigste Aufgabe des Internet Protocol ist es, einen Weg durch das verzweigte Netzwerk des Internets zu finden und somit die Daten vom Sender zum Ziel weiterzuleiten. Hierzu stellt das Protokoll einen ungesicherten, verbindungslosen Dienst bereit. Demnach existiert keine Garantie, dass die Daten ihr Ziel erreichen. In der Regel werden die Daten zum Transport in verschiedene Datenpakete unterteilt. Datenpakete können verloren gehen oder in unterschiedlicher Reihenfolge am Ziel eintreffen. Problematisch ist der Verlust eines einzelnen Datenpaketes, da das Internet Protocol dies als Gesamtverlust wertet. Für diesen Transsport wird zu den eigentlichen Daten ein Header hinzugefügt. Dieser besteht aus Informationen wie der Ziel-IP-Adresse, der Paketlänge in Bytes, einer Prüfsumme sowie weiteren Informationen. [7]

Das Transmission Control Protocol TCP baut auf dem Internet Protocol auf. Es realisiert eine verbindungsorientierte Kommunikation zwischen verteilten Rechnern. Dieses Transportprotokoll ermöglicht eine Fehler- sowie eine Flusskontrolle. Daher können die Daten, die das Ziel in unterschiedlicher Reihenfolge erreichen, wieder korrekt zugeordnet werden. Hierzu nummeriert TCP die Datenpakete und arrangiert somit einen zuverlässigen Datentransport.

Ein weiteres Protokoll ist das User Datagram Protcol UDP, das ebenfalls auf dem Internet Protocol aufbaut, aber im Gegensatz zu TCP eine verbindungslose Kommunikation realisiert. Hierbei erfolgt keine Fehler- und Flusskontrolle. Aus diesem Grund ist kein zuverlässiger Datentransport gegeben, allerdings müssen auch weniger Informationen übertragen werden. Demnach ist der UDP Header geringer als der TCP Header und daher ist eine schnellere Datenübertragung möglich.

Sowohl UDP als auch TCP benutzen sogenannte Ports, um verschiedene Anwendungen unterscheiden zu können. Somit kann eine einzelne IP-Adresse mehrere Anwendungen wie zum Beispiel einen Webbrowser und einen E-Mail-Dienst gleichzeitig benutzen. Die Angabe der Portnummer ist im Header beider Protokolle 16 Bit lang, daher können bis zu 65535 Ports gleichzeitig geöffnet werden. [8]

2.1.4 Referenzmodell und Anwendungsbeispiele

Das Dynamic Host Configuration Protocol DHCP ordnet neuen Clients in einem Netzwerk eine IP-Adresse aus einem vorgegeben Adressraum sowie weitere Konfigurationsparameter zu. Ein DHCP-Server ist ein Rechner, der alle Konfigurationsparameter für die Clients gespeichert hat.

Zum Anmelden bei einem Server fordert der DHCP-Client eine IP-Adresse und die weiteren Konfigurationsparameter an. [1]

	OSI-Referenzmodell	TCP-IP-Protokolle
Schicht 7	Anwendungsschicht	HTTP, DHCP, DNS, …
Schicht 6	Darstellungsschicht	-
Schicht 5	Kommunikationsschicht	-
Schicht 4	Transportschicht	TCP, UDP
Schicht 3	Vermittlungsschicht	IP
Schicht 2	Sicherungsschicht	ARP, Ethernet, WLAN, …
Schicht 1	Physikalische Schicht	Bildet mit Schicht 2 die Netzzugangsschicht

Abbildung 2-4 - Vergleich OSI-Modell und IP-Modell

In der Abbildung 2-4 sind die Protokolle des Internets im Vergleich zum OSI-Referenzmodell aufgelistet. Die Schichten eins und zwei bilden zusammen die Netzzugangsschicht. Die erläuterten Protokolle ARP, IP, TCP und UDP sind den jeweiligen Schichten zugeordnet. Außerdem sind auf Schicht sieben Anwendungsbeispiele wie beispielsweise HTTP, DHCP und DNS genannt.

Durch Verwendung des Domain Name Service DNS wird eine Auflösung von Webadressen in IP-Adressen und umgekehrt ermöglicht. Auf Grund des Dienstes kann im Webbrowser eine URL statt einer IP-Adresse eingegeben werden. DNS vereinfacht die Adressierung an den Zielserver. [5]

2.1.5 Netzwerkverschlüsselung

Die Verschlüsselung von Nachrichten ist notwendig, da in Netzwerken mehrere Teilnehmer die gleiche Infrastruktur benutzen. Besitzt eine dritte Person ebenfalls Zugang zu der gleichen Infrastruktur wie beispielsweise einem WLAN, kann diese den Datenverkehr zwischen den Teilnehmern mitlesen. Da die übertragenen Informationen private und sensible Daten enthalten,

muss der Datenverkehr verschlüsselt werden. Das WLAN wird in den meisten Fällen mit einem gemeinsamen Geheimnis, dem Passwort, verschlüsselt. Sobald ein Passwort für das WLAN festgelegt ist, werden die Daten nicht mehr im Klartext übertragen. Bevor der Teilnehmer die Daten zum Router versendet, verschlüsselt er die Daten mit dem Passwort. Danach werden die Daten, wie in einem Tresor verschlossen, über das WLAN zum Router übertragen. Der Router kann die Daten dennoch lesen, da er ebenfalls das Passwort für den Tresor kennt. Eine dritte Person, die das Passwort für diesen Tresor nicht kennt, kann nur den Tresor, aber nicht den Inhalt sehen.

Bis in die 1970er Jahre wurden symmetrische Verfahren zu Verschlüsslung benutzt. Bei diesen benutzen der Sender und der Empfänger den gleichen Schlüssel für die Ver- und Entschlüsselung. Problematisch war der Austausch des gemeinsamen Schlüssels, da diese Übertragung abgehört werden konnte. Danach wurden asymmetrische Verfahren, bei denen ein Schlüsselpaar aus einem öffentlichen und einem privaten Schlüssel verwendet wird, etabliert. Wie aus den Bezeichnungen hervorgeht, ist nur der private Schlüssel geheim. Aufgrund dieses Verfahrens muss kein geheimer Schlüssel mehr übertragen zu werden. Zum Versenden von Daten an einen Teilnehmer wird sein öffentlich bekannte Schlüssel verwendet. So kann ohne die Übertragung des geheimen Schlüssels eine verschlüsselte Kommunikation zwischen Netzwerkteilnehmern etabliert werden. [9]

2.1.6 Verschlüsselung von WLAN-Netzwerken

Der erste Standard, um WLAN Netzwerke zu verschlüsseln, war Wired Equivalent Privacy WEP. Dieser Standard gilt inzwischen als nicht mehr sicher, daher wurde der Standard WiFi-Protected-Access WPA eingeführt. Damit ältere Hardware weiterhin unterstützt wurde, wurden zwei unterschiedliche Verschlüsselungsstandards eingeführt. Das Temporal Key Integrity Protocol TKIP sollte die ältere Hardware unterstützen. Der zweite Standard ist der Advanced Encryption Standard - Counter-Mode/CBC-MAC Protocol AES-CCMP. Geräte, die TKIP unterstützen und nicht zwangsläufig AES-CCMP unterstützen, benutzen WPA Verschlüsselung. Unterstützt ein Gerät AES-CCMP und TKIP ist optional, dann handelt es sich um den aktuellsten Standard WPA2. Die meisten privaten Netzwerke verwenden bei der Authentifizierung ein gemeinsames Geheimnis und damit ein symmetrisches Verschlüsselungsverfahren. Dieses Geheimnis ist in den meisten Fällen ein Passwort. Diese Methode wird Pre Shared Key PSK genannt. Der Schlüssel wird auch als Pairwise Master Key PMK bezeichnet. Zur Anmeldung in ein Netzwerk ist ein 4-Wege-Handshake nötig.

Der Access Point sendet einen Nonce-Wert an den Teilnehmer. Ein Nonce-Wert ist eine nur einmal verwendete Buchstaben- oder Zahlenkombination. Daraufhin sendet der Teilnehmer einen Nonce-Wert zurück. Der Access Point sendet einen verschlüsselten Schlüssel an den Teilnehmer, der zur Sicherung von Broadcast-Nachrichten verwendet wird. Der Teilnehmer bestätigt abschließend dem Empfang.

Sowohl der Access Point als auch der Teilnehmer berechnen aufgrund der Nonce-Werte, dem PMK und den MAC-Adressen einen Pairwise Transient Key PTK. Dieser wird aufgeteilt in einen Schlüssel für das TKIP- bzw. CCMP-Verfahren und zwei Schlüssel für die Dauer des Handshakes. Aufgrund dieses Verfahrens ist es für einen unberechtigten Teilnehmer nicht möglich den PMK zu berechnen. Ein Angreifer hat nur die Möglichkeit bei einer PSK Verschlüsselung mit einer Wörterbuch- bzw. Brute-Force-Attacke das Passwort herauszufinden. [10]

2.1.7 Message Queue Telemetry Transport MQTT

Im Internet der Dinge wird für den Datenaustausch zwischen Sensoren und Aktoren ein Protokoll mit verschiedenen Anforderungen benötigt. Viele Geräte müssen autark versorgt werden, daher ist ein geringer Stromverbrauch wesentlich. Außerdem wächst die Anzahl der vernetzten Geräte rasant, sodass ein mögliches Protokoll eine gute Skalierbarkeit aufweisen muss. Das auf einen geringen Overhead und eine gute Skalierbarkeit ausgelegt MQTT Protokoll erfüllt diese Anforderungen. Der Overhead bezeichnet die Informationen, die das Protokoll den Nutzdaten für die korrekte Übertragung hinzufügt. MQTT funktioniert auch in Netzen mit Verzögerungen und geringem Durchsatz. Durch diese Eigenschaften werden sowohl Batterie- als auch Übertragungsressourcen geschont. Üblicherweise benutzt MQTT den TCP- bzw. UDP-Port 1883. Eine weitere Besonderheit des Protokolls ist die Entkopplung von Sender und Empfänger. Der zentrale Server, der sogenannte Broker, empfängt alle Nachrichten und leitet diese an die Empfänger weiter. Der Sender überträgt die Daten unter einem bestimmten Thema an den Broker. Dieser leitet diese Nachricht an jeden Empfänger, der sich vorher beim Broker für dieses Thema angemeldet hat, weiter. Außerdem besteht die Möglichkeit, dass der Broker die Nachrichten speichert, sollte ein Empfänger nicht erreichbar sein. Diese Zuverlässigkeit in der Nachrichtenzustellung kann durch die unterschiedlichen Quality-of-Service-Level QoS konfiguriert werden. Es besteht die Möglichkeit zwischen keiner Zustellungsgarantie, mindestens eine Zustellung oder genau eine Zustellung zu wählen. [11]

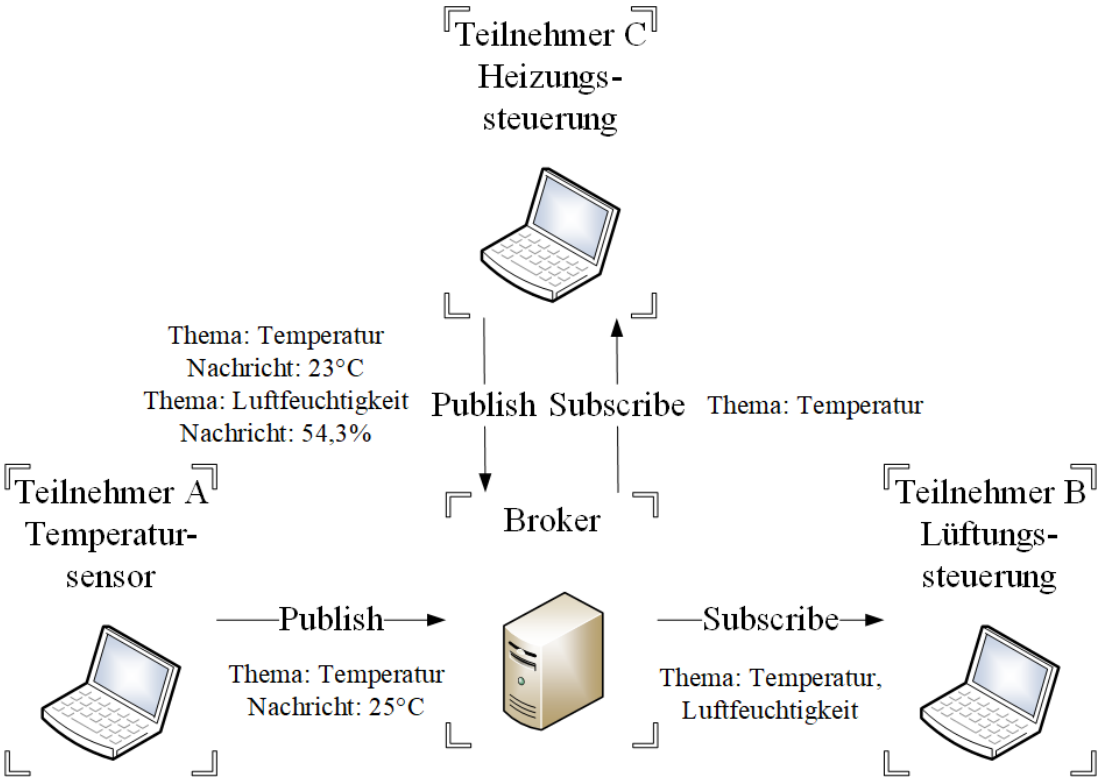

Abbildung 2-5 - Anwendungsbeispiel des MQTT-Protokolls

In der Abbildung 2-5 wird das MQTT-Protokoll am Beispiel einer Hausautomation dargestellt. In diesem Fall gibt es drei Teilnehmer A, B und C. Der Temperatursensor sendet die Nachricht „25°C“ unter dem Thema „Temperatur“ an den Broker. Das Senden von Nachrichten wird als Publishen bezeichnet. Der Broker leitet die Nachricht an alle Teilnehmer weiter, die das Thema „Temperatur“ abonniert haben. In diesem Fall haben die Lüftungs- und die Heizungssteuerung das Thema abonniert. Das Abonnieren von Themen wird als Subscribe bezeichnet. Außerdem misst die Heizungssteuerung die Temperatur und die Luftfeuchtigkeit. Die Heizungssteuerung sendet diese Informationen ebenfalls unter dem jeweiligen Thema und hat gleichzeitig das Thema Temperatur abonniert. Es können also mehrere Teilnehmer unter dem gleichen Thema Nachrichten an den Broker senden. Dieser verteilt die Nachrichten an die Abonnenten. Die Lüftungs- und Heizungssteuerung können somit in Abhängigkeit der Temperatur und der Luftfeuchtigkeit ein angenehmes Raumklima einstellen. Sollte der Temperatursensor ausfallen, erhält die Lüftungssteuerung immer noch die Daten von der Heizungssteuerung. Ein Ausfall des Brokers bedeutet allerdings einen Systemausfall, da der Broker den zentralen Baustein des Protokolls darstellt.

2.2 Schwachstellen in der Datensicherheit von Rechnernetzwerken

Durch die steigende Vernetzung von einzelnen Rechnern, können immer mehr Daten ausgetauscht werden. Der internationale Datenaustausch wird rasant beschleunigt und Nachrichten verbreiteten sich in wenigen Sekunden. Vor allem für internationale Unternehmen und Forschungsgruppen ist die vereinfachte Kommunikation eine große Erleichterung. Allerdings werden hierbei auch sensible und interne Dokumente ausgetauscht. Durch umfangreiche Netzwerkkenntnisse gelingt es Unbefugten immer wieder geheime Dokumente mitzulesen und in Firmennetzwerke einzudringen. Die entstehenden Schäden sind ein Problem für die Unternehmen. Die Dokumente müssen verschlüsselt, geschützt und versteckt werden. Außerdem müssen eigene Mitarbeiter geschult werden, um Angriffe zu vermeiden und Firmennetzwerke zu schützen.

Im Zuge der Digitalisierung werden immer mehr Haushalte durch internetfähige Geräte vernetzt, die sich über das Internet automatisiert steuern lassen. Aufgrund der Entwicklung zum Internet der Dinge müssen sich somit auch Privatpersonen zunehmend mit dem Thema Datensicherheit auseinandersetzen.

2.2.1 WLAN Adapter

Der WLAN Adapter übernimmt die Aufgabe Datenpakete zu empfangen und zu versenden. In den meisten Fällen wird der Adapter vom Betriebssystem verwaltet und somit der Betriebsmodus bestimmt. Im üblichen Betriebsmodus leitet der Adapter nur an ihn adressierte Datenpakete an das Betriebssystem und die Anwendungen weiter. Ein anderer Betriebsmodus, der sogenannten Monitormodus, ermöglicht es, alle Datenpakete zu empfangen und in den Anwendungen zu verarbeiten. Somit können auch Datenpakete anderer WLANs mitgeschnitten werden. Außerdem besteht die Möglichkeit den 4-Wege-Handshake aufzuzeichnen und eine Wörterbuch- bzw. Brute-Force-Attacke vorzubereiten. Im bekannteren Promiscuous-Modus können nur Datenpakete im angemeldeten Netzwerk mitgeschnitten werden. Der Monitormodus ist daher noch freizügiger als der Promiscuous-Modus. [12]

2.2.2 Social Engineering

Social Engineering befasst sich mit dem Menschen als Schwachstelle in einem System, anstatt mit dem System selbst. Hierbei sollen über die Schwachstelle Mensch Passwörter ausgespäht, Daten gesammelt oder Profite erwirtschaftet werden. Die Angriffsmöglichkeiten sind sehr vielfältig und es muss auf immer neuere Bedrohungen reagiert werden können. Die Methoden gehen von Telefonanrufen von vermeintlichen Systemadministratoren über Mails von

Geschäftskollegen mit enthaltenen Links hin zu schädlichen USB Sticks, die auf dem Firmengelände verstreut werden. In den meisten Fällen wird die Hilfsbereitschaft der Opfer ausgenutzt.

Das bekannteste Beispiel ist das sogenannte Phishing. Hierbei sollen Passwörter ausgespäht werden. Dadurch erlangen die Angreifer an wichtige Ressourcen wie beispielsweise firmeninterne Dokumente.

Auf Grund der sehr effektiven Angriffe müssen Sicherheitsbeauftragte der Firmen mit rigorosen Maßnahmen durchgreifen. Diese reichen von obligatorischen Sicherheitsschulungen über strenge Filterung von E-Mails und Internetseiten bis hin zur Sperrung von USB-Ports an Arbeitsplatzrechnern.

Ebenso sind immer mehr Privatpersonen von Social-Engineering-Attacken betroffen. Häufig werden über E-Mails vermeintliche Gewinne ausgeschüttet oder gefälschte Mahnungen von Firmen versendet. Diese Täuschungen zielen allesamt auf die Schwachstelle Mensch ab. Wird der enthaltene Link angeklickt oder die Datei geöffnet, ist der Computer infiziert. [13]

2.2.3 ARP-Spoofing

Das in Abschnitt 2.1.2 erläuterte Address Resolution Protocol ARP weist eine Sicherheitslücke auf. Das sogenannte ARP-Spoofing oder auch ARP-Cache-Poisoning ist die Grundlage für einen sogenannten Man-in-the-Middle-Angriff . Befindet sich der Angreifer bereits in dem Netzwerk, kann er eine gefälschte Adressantwort versenden. Er kann im Namen von einem anderen Teilnehmer eine Adressantwort mit seiner MAC-Adresse versenden. Die meisten Betriebssysteme übernehmen keine Überprüfungen, ob vorher eine Adressanfrage gesendet wurde. Damit werden die an einen bestimmten Teilnehmer gesendeten Datenpakete über den Angreifer gelenkt. Dieser leitet die Datenpakete direkt weiter an das jeweilige Ziel. Mit dieser Methode besteht für den Angreifer die Möglichkeit sämtliche Datenpakete mitzulesen und zu manipulieren. Da der Angreifer alle Datenpakete direkt weiterleitet, fällt den angegriffenen Teilnehmern der Mittelsmann nicht auf. Aufgrund dieser unauffälligen Methode kann über lange Zeiträume der Datenverkehr unbemerkt abgehört werden. [13]

2.2.4 Man-in-the-Middle

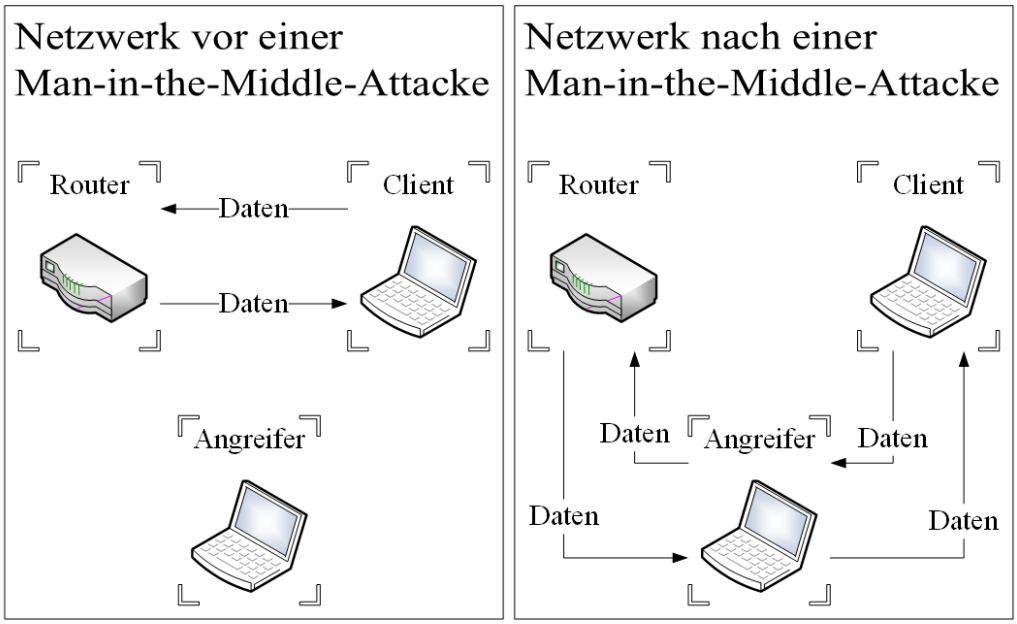

Abbildung 2-6 - Netzwerkverkehr mit und ohne Man-in-the-Middle-Attacke

In der Abbildung 2-6 ist der Verkehr in einem Netzwerk mit und ohne eine Man-in-the-Middle-Attacke dargestellt. In einem normalen Netzwerk kann der Client seine Daten direkt mit dem Router austauschen und der Angreifer kann diese Daten nicht manipulieren. Allerdings kann der Angreifer die Adressauflösung durch das ARP-Protokoll mit dem im Abschnitt 2.2.3 ARP-Spoofing manipulieren, damit der Client die Daten, die für den Router bestimmt sind, zuerst an den Angreifer sendet. Dieser leitet die Daten dann weiter an den Router. Auch der Router sendet die Daten für den Client zuerst über den Angreifer und dieser leitet sie weiter. Durch diesen Angriff kann der Angreifer die Daten mitlesen, analysieren und auch manipulieren. Durch die direkte Weiterleitung erhalten sowohl der Router als auch der Client alle Daten und bemerken den Angreifer nicht. Lediglich durch eine Analyse der ARP-Cache kann dieser Angriff aufgedeckt werden.

2.2.5 Denial of Service

Durch Denial-of-Service- DoS bzw. Distributed-Denial-of-Service-Angriffe DDoS können Zielsysteme wie Server und Netzwerke lahmgelegt werden. Bei einem DDoS- wird im Gegensatz zum DoS-Angriff die Überlastung der Infrastruktur durch einen Verbund mehrerer Rechner erzeugt. Außerdem hat ein DDoS-Angriff den Vorteil, dass der eigentliche Angreifer deutlich schwerer zu identifizieren ist. Hierzu müssen die Rechner bereits vorher infiziert werden, um für den Angriff zur Verfügung zu stehen. Das so entstandene Rechnernetz wird Botnetz genannt. Die meisten Angriffe lassen sich auf Vandalismus, Sabotage oder Erpressung

zurückführen. Durch den aktuellen Trend alle Geräte, häufig unter mangelnder Beachtung von Sicherheitsaspekten, zu vernetzen, entstehen mächtige Botnetze. Diese können im Internet angemietet und somit sehr leicht große Rechensysteme angegriffen werden.

In der Abbildung 2-7 wird die Struktur eines solchen Angriffs dargestellt. Der Angreifer kontrolliert über einen Server das Botnetz. In diesem Fall besteht das Botnetz aus vier Teilnehmern. In der Realität bestehen Botnetze aus mehreren Millionen Teilnehmern. Sobald der Kontrollserver das Signal gibt, können diese Verbindungen zu einem Webserver aufbauen und ihn damit überlasten. Der eigentliche Angreifer ist schwer zu identifizieren, da der Webserver nur Anfragen von Geräten aus dem Botnetz erhält.

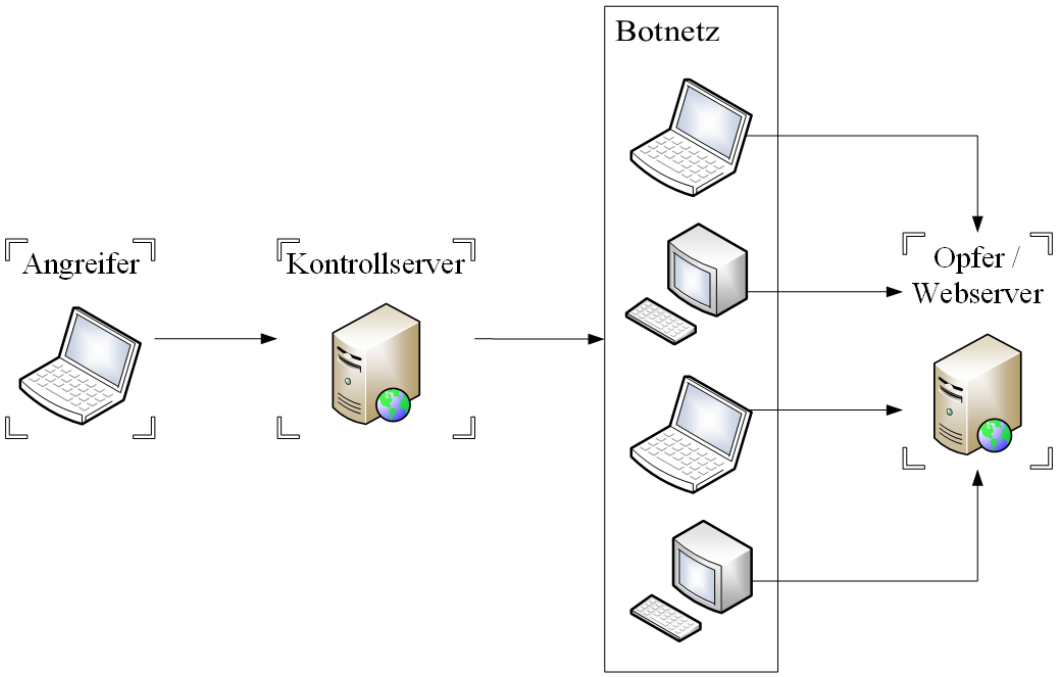

Abbildung 2-7 - Darstellung DDoS-Angriff

Es wird zwischen drei Angriffsarten unterschieden. Den Low-Level Protokoll-Angriffen, dem HTTP-Flooding und den DNS Attacken. In vielen Fällen werden diese Methoden kombiniert eingesetzt.

Die Low-Level Protokoll-Angriffe basieren auf riesigen Datenmengen, die den Netzwerkverkehr überschwemmen und dabei Protokolle der unteren Schichten des OSI-Referenzmodells verwenden. Häufig werden auch Schwachstellen in den Protokollen ausgenutzt. Zum Beispiel wird bei dem Aufbau einer TCP-Verbindung vom Client ein Datenpaket mit SYN-Flag gesendet. Der Router antwortet mit einem SYN-ACK-Paket. Um den Vorgang abzuschließen muss der Client nun ein ACK-Paket senden. Bei einem Angriff werden tausende Datenpakete mit einem SYN-Flag gesendet, aber kein zum Abschluss nötiges

Datenpaket mit ACK-Flag. Dadurch hat der Server tausende offene Verbindungen und kann somit überlastet werden. Dieser Angriff wird SYN-Flooding genannt.

Beim HTTP-Flooding werden herkömmliche Anfragen an die Webserver gestellt, deshalb ist dieser Angriff sehr schwer vom normalen Datenverkehr zu unterscheiden. Auch hier werden verschiedene Techniken eingesetzt. Beispielsweise sendet der Angreifer eine Anfrage an die Startwebsite, welche als Antwort auch ihre Unterseiten mitsendet. Durch die Analyse der Antwort können auch alle Unterseiten mit Anfragen attackiert werden. Auch die durch Teilanfragen erzeugten offenen Verbindungen können den Webserver schnell überlasten.

Das Domain Name System DNS übersetzt den Websitenamen in eine numerische IP-Adresse. Ein Ausfall dieses Systems sorgt dafür, dass die Website nicht mehr unter ihrem Namen erreichbar ist. Eine Möglichkeit ist es, den DNS-Server mit fehlerhaften UDP-Datenpaketen zu attackieren. Dieser wird versuchen, die Datenpakete zu beantworten und zu validieren, dabei steigt die Belastung für Hardware- und Betriebssystemressourcen. Häufig kann der Server so zu einem Neustart gezwungen werden und steht nicht mehr für Anfragen zur Verfügung. [14]

2.2.6 WLAN Managementfunktionen

Im WLAN werden nicht alle Daten verschlüsselt übertragen. Sowohl beim Anmelde- als auch beim Abmeldevorgang werden die sogenannten Management Frames nicht verschlüsselt, da zu diesem Zeitpunkt die verschlüsselte Kommunikation erst noch etabliert werden muss. Der Anmeldevorgang besteht aus der Authentication anschließend der Association. Beim Abmelden wird lediglich ein De-Authentication Paket übertragen. Durch die unverschlüsselte Übertragung bieten sich zahlreiche Angriffsmöglichkeiten auf das Zielsystem. Alle drei Varianten blockieren die Verfügbarkeit eines Dienstes, sind also den Denial-of-Service-Angriffen zuzuordnen.

Beim Authentication-Flooding wird eine große Anzahl von Authentication Frames an den Access Point gesendet. Durch dieses Verfahren kann ein unzureichend geschützter Access Point den Dienst verweigern, da er nur eine gewisse Anzahl von Anfragen pro Zeiteinheit abarbeiten kann.

Das Association-Flooding hingegen simuliert durch gefälschte MAC-Adressen den parallelen Zugriff vieler Clients. Jeder Client erhält bei der Association eine eigene ID. Dieser Angriff zielt darauf ab, alle verfügbaren IDs zu blockieren, um die eigentlichen Clients am Anmelden im Netzwerk zu hindern.

Zum Abmelden können Clients durch das Deauthentication-Flooding gezwungen werden, entweder erhält der Client mit der MAC-Adresse des Access Points ein Deauthentication Paket

oder anders herum. Durch diese Methode kann die Kommunikation im WLAN lahmgelegt werden oder die Clients zu einem erneuten Login gezwungen werden. [15]

2.2.7 Exploits

Durch Exploits werden Sicherheitslücken ausgenutzt, um sich Zugriff zum System zu verschaffen. Diese Schwachstellen sind meisten Implementierungsfehler in Programmen oder Betriebssystemen. Durch Skripte werden die betroffenen Stellen attackiert und der Angreifer kann die Kontrolle über das Betriebssystem übernehmen, Daten auslesen, einen dauerhaften Fernzugriff einrichten oder weitere Schadsoftware auf den Zielrechner herunterladen. Ein dauerhafter Fernzugriff wird meistens über einen selten genutzten TCP- oder UDP-Port realisiert. Ein im Hintergrund laufendes Programm öffnet auf einem dieser Ports eine Verbindung, die der Angreifer bei Bedarf aktivieren kann. Der Schad-Code, der auf dem Zielrechner ausgeführt werden soll, wird als Payload bezeichnet. Nachdem die Schwachstellen bekannt werden, werden diese meistens durch Updates geschlossen. Häufig ist das Betriebssystem des Opfers nicht bekannt. In diesen Fällen ist es wichtig, dass ein plattformunabhängiger Payload generiert wird. Wird ein plattformabhängiger Payload verwendet, kann das Opfer möglichweise die erstellte Schadsoftware nicht ausführen. Durch Programmiersprachen wie beispielsweise JavaScript oder Python kann ein plattformunabhängiger Payload erstellt werden.

Um einen Exploit zu generieren, stehen zahlreiche Exploit-Kits zur Verfügung. Diese greifen auf Datenbanken, in denen bekannte Sicherheitslücken gespeichert werden, zurück. So werden beispielsweise PDF Dateien oder ausführbare Programme erstellt, die Sicherheitslücken attackieren können. Diese müssen dann auf dem Zielrechner geöffnet bzw. ausgeführt werden. Häufig werden über Exploits versteckte Hintereingänge zu Rechnern erschaffen. Diese werden dann bei Bedarf für eine DDoS Attacke aktiviert. [16]

2.3 Künstliche Intelligenz

Der Begriff künstliche Intelligenz KI definiert ein Teilgebiet der Informatik, bei dem Aufgaben möglichst effizient gelöst werden sollen und dabei die menschliche Vorgehensweise der Problemlösung nachgebildet werden soll. Der Überbegriff wird in weitere Teilgebiete wie zum Beispiel maschinelles Lernen unterteilt. [17]

2.3.1 Agenten

In der Literatur wird im Zusammenhang mit künstlicher Intelligenz häufig der Ausdruck Agent verwendet. Dieser Begriff bezeichnet ein Programm, das im Auftrag eines anderen Programms oder eines Menschen selbstständig Aufgaben erledigt. Agenten werden nach verschiedenen Anforderungen charakterisiert und bekommen Eigenschaften wie zum Beispiel autonom, lernend oder reaktiv zugewiesen. Des Weiteren werden reine Software-Agenten, welche aus Benutzereingaben Ergebnisse berechnen und Hardware-Agenten, welche zusätzliche über Sensoren und Aktoren verfügen, unterschieden. [18]

2.3.2 Maschinelles Lernen

Ein wichtiger Aspekt in der menschlichen Vorgehensweise bei der Problemlösung ist der Lernprozess. Bislang sind Maschinen dem Menschen im Bereich Lernen deutlich unterlegen, daher beschäftigt sich das Teilgebiet maschinelles Lernen mit diesem Prozess. Durch die steigende Komplexität im Bereich der künstlichen Intelligenz werden bereits verschiedene Lernverfahren eingesetzt. Häufig kommt es auch zu einer Mischung aus programmiertem und gelerntem Verhalten.

Das Auswendiglernen ist für Maschinen deutlich leichter als für Menschen. Durch simples Speichern von Dateien kann eine Maschine bereits den Text beliebig oft wiedergeben und dauerhaft archivieren. Interessanter ist das Erlernen von Mustern und Techniken. Ein Computer kann die Lösungen beliebig vieler Additionen kennen und immer die richtige Antwort auf genau diese Additionen ausgeben. Wird der Computer allerdings mit einer unbekannten Addition konfrontiert, so fehlt ihm die Lösung. Deswegen ist das Ziel maschinellen Lernens durch Analyse von Trainingsdaten die Logik hinter den Operationen zu erlernen. Dieser Vorgang wird als Generalisierung bezeichnet. Demnach soll die mathematische Operation der Addition erlernt und auf weitere unbekannte Daten angewandt werden können.

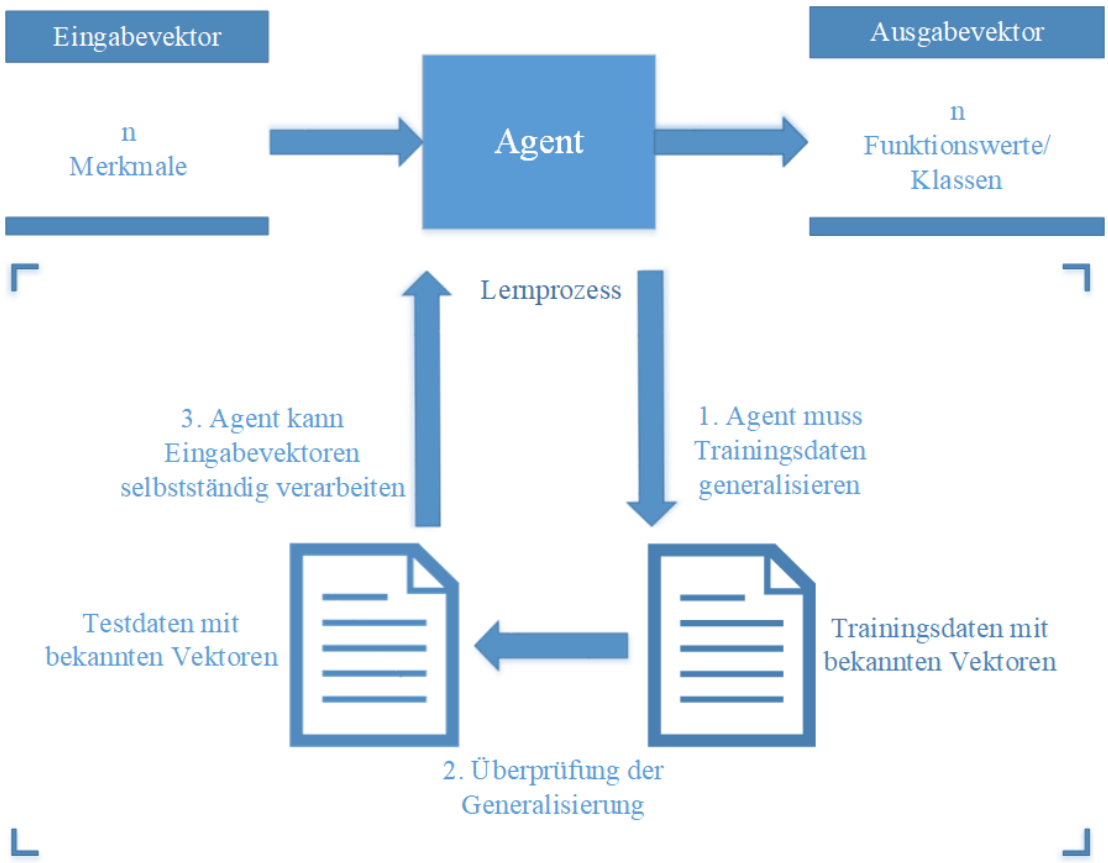

Abbildung 2-8 - Prinzip des maschinellen Lernens

In der Abbildung 2-8 ist ein lernender Agent dargestellt. Dieser berechnet aus beliebig vielen Merkmalen über eine Funktion, welche durch Trainingsdaten angelernt wird, einen oder mehrere Ausgabewerte. Je nach Anwendungsfall kann die Bestimmung der Ausgabewerte durch eine Regression oder eine Klassifizierung erfolgen. Die erlernte Funktion gilt auch für unbekannte Daten. Um die Generalisierungsfähigkeit des Programms zu testen, wird der Agent mit für ihn unbekannten Testdaten konfrontiert und die berechneten mit den bekannten Ergebnissen verglichen. Die korrekt berechneten Ergebnisse werden als Leistungsindex in Prozent angegeben. Demnach ist das Ziel maschinellen Lernens, einen bestmöglichen Funktionsapproximator zu erhalten. Um den Agenten zu trainieren und dabei die gewünschte Generalisierung zu erreichen gibt es unterschiedliche Lernverfahren.

Das überwachte Lernen, welches auch Lernen mit Lehrer genannt wird, hat sich bereits etabliert. Ein Lehrer stellt zu jedem Eingabevektor den passenden Ausgabevektor zur Verfügung und demnach kann über Verfahren wie Entscheidungsbäume, neuronale Netze oder Bayer-Netze die gewünschte Funktion antrainiert werden.

Das Gegenteil vom überwachten Lernen ist das Lernen ohne Lehrer bzw. nicht überwachtes Lernen. Bei diesem Verfahren werden sogenannte Clusteringverfahren eingesetzt, um für einen Eingabevektor ein Modell zu generieren und somit Vorhersagen zu erstellen, da der

Ausgabevektor nicht bekannt ist. Dieses Verfahren wird bei Suchmaschinen eingesetzt, um Ähnlichkeiten zur Eingabe festzustellen und diese auszugeben.

Ein noch sehr junges Teilgebiet des maschinellen Lernens ist das teilüberwachte Lernen. Bei diesem haben nur wenige der vielen Trainingsdaten einen bekannten Ausgabevektor.

Ein weiteres Lernverfahren ist das Lernen durch Bestärkung. Durch Belohnung und Bestrafung soll bei diesem Verfahren dem Agenten eine bestmögliche Funktion antrainiert werden.

Im Bereich der automatischen Merkmalsextraktion wird noch intensive Forschung betrieben. Die meisten Verfahren benötigen vorbereitete Datensätze, um die Funktionen zu erlernen. Ziel dieser Forschung ist es, direkt aus den Rohdaten die wichtigsten Merkmale zur Berechnung der Funktionen bestimmen zu können. Das sogenannte Deep Learning benutzt hierzu komplexe neuronale Netze zusammen mit Vorverarbeitung ohne Lehrer. Auf diesem jungen Forschungsgebiet konnten erfolgreiche Algorithmen entwickelt werden, daher erlebt der Begriff des maschinellen Lernens einen erneuten Medienhype. [18]

In der Abbildung 2-9 ist dieser Vorteil nochmals bildlich verdeutlicht. Der manuelle Aufwand, die Merkmale auszuarbeiten, fällt hierbei weg. Die Eingabe sei, ein beliebiges Bild mit der Aufgabe zu identifizieren, ob sich eine Sonne auf dem Bild befindet oder nicht. Der genannte Vorteil von Deep Learning wird direkt sichtbar.

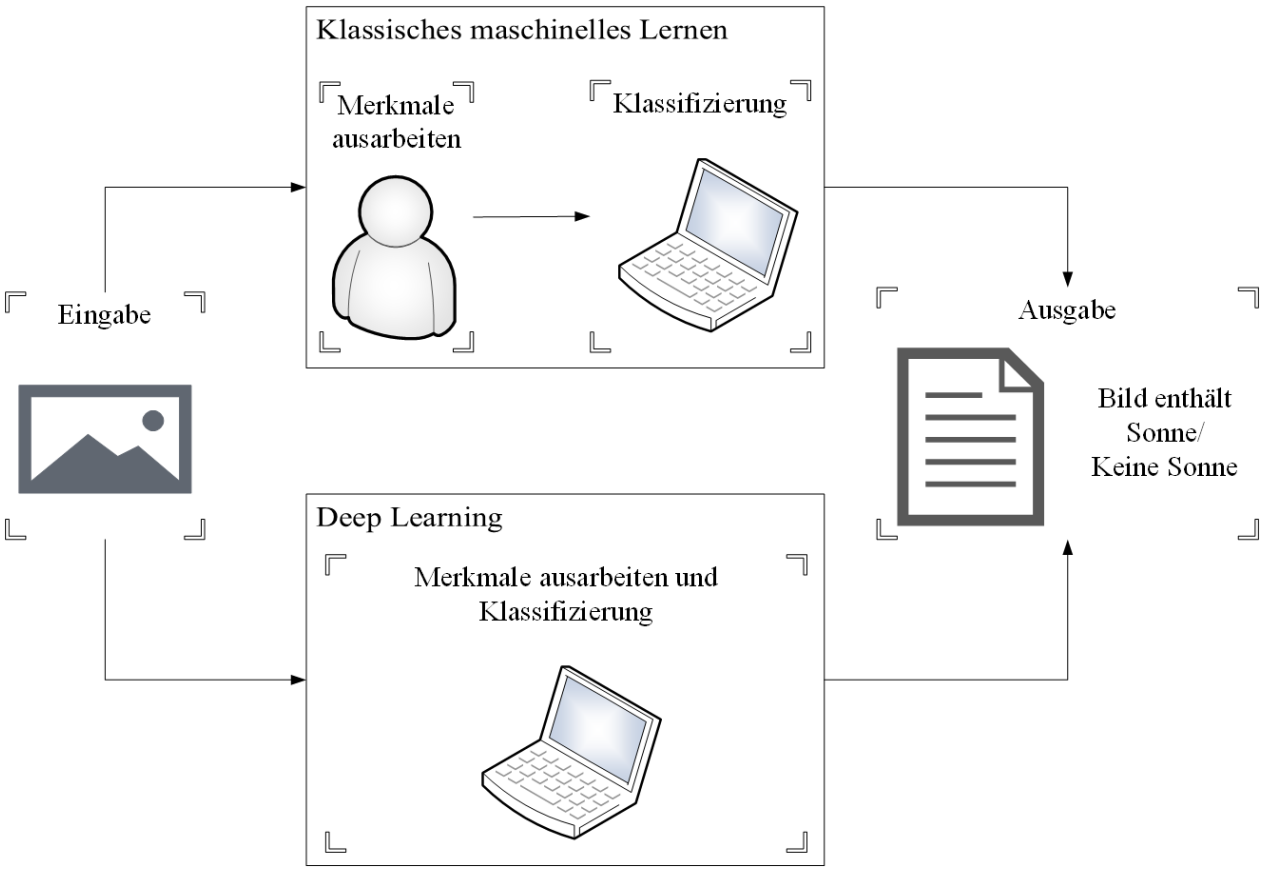

Abbildung 2-9 - Vorteil von Deep Learning am Beispiel Bilderkennung

2.3.3 Neuronale Netze

Der Begriff der neuronalen Netze ist aus der Biologie bekannt. Er beschreibt ein Netzwerk aus Nervenzellen im Gehirn von Menschen und Tieren. Der Zellkörper eines Neurons entspricht vereinfacht ausgedrückt einem Speicher für Spannungsimpulse. Diese Impulse kommen von anderen verknüpften Neuronen. Nachdem ein bestimmter Schwellwert überschritten wurde, löst das Neuron aus und der Speicher wird entladen. Daraufhin erreicht dieser Spannungsimpuls andere Neuronen. Die Neuronen sind untereinander über Synapsen verbunden. Es gibt keinen Schaltplan eines menschlichen Gehirns und es ist auch nicht erstrebenswert, denn diese Struktur des Netzwerkes aus Neuronen ist adaptiv. Die Synapsen sind der adaptive Anteil, denn je öfter die Spannungsimpulse übertragen müssen, desto höher wird ihre Leitfähigkeit. Im Gegensatz dazu sinkt die Leitfähigkeit von kaum benutzen Synapsen. Dieser Vorgang kann sogar das Absterben von Synapsen zur Folge haben. Demnach passt sich die Struktur ihrer Belastung an. Es ist noch immer ungeklärt, wie durch diesen adaptiven Prozess ein intelligentes Verhalten erreicht wird.

Das Verhalten wird im mathematischen Modell von künstlich neuronalen Netzen modelliert. Neuronale Netze werden dem Teilbereich des maschinellen Lernens innerhalb der künstlichen Intelligenz zugeordnet. Jedes Neuron hat in vielen Strukturen alle Ausgänge der Neuronen der vorherigen Schicht als Eingangsparameter. Das Neuron bildet die Summe aller Eingangsparameter, wobei jeder Eingangsparameter noch mit einem individuellen Faktor multipliziert wird. Häufig wird noch ein Offset auf die Summe addiert. Anschließend wird auf das Ergebnis eine Aktivierungsfunktion angewendet. Diese simuliert den Zusammenhang zwischen Eingangsparameter und Aktivitätslevel eines Neurons. Es werden anwendungsabhängig verschiedene Aktivierungsfunktionen verwendet. Besonders wichtig ist hierbei, ob der Ausgangsparameter stetig oder unstetig sein soll. Außerdem können mit verschiedenen Aktivierungsfunktionen die Ergebnisse der Berechnung verbessert werden. Für stetige Ausgangsparameter werden häufig die Sigmoidfunktion, der hyperbolische Tangens oder Funktionen mit verschiedenen linearen Steigungen verwendet. Hingegen wird bei unstetigen Ausgangsparameter meisten die Sprungfunktion benutzt. [18]

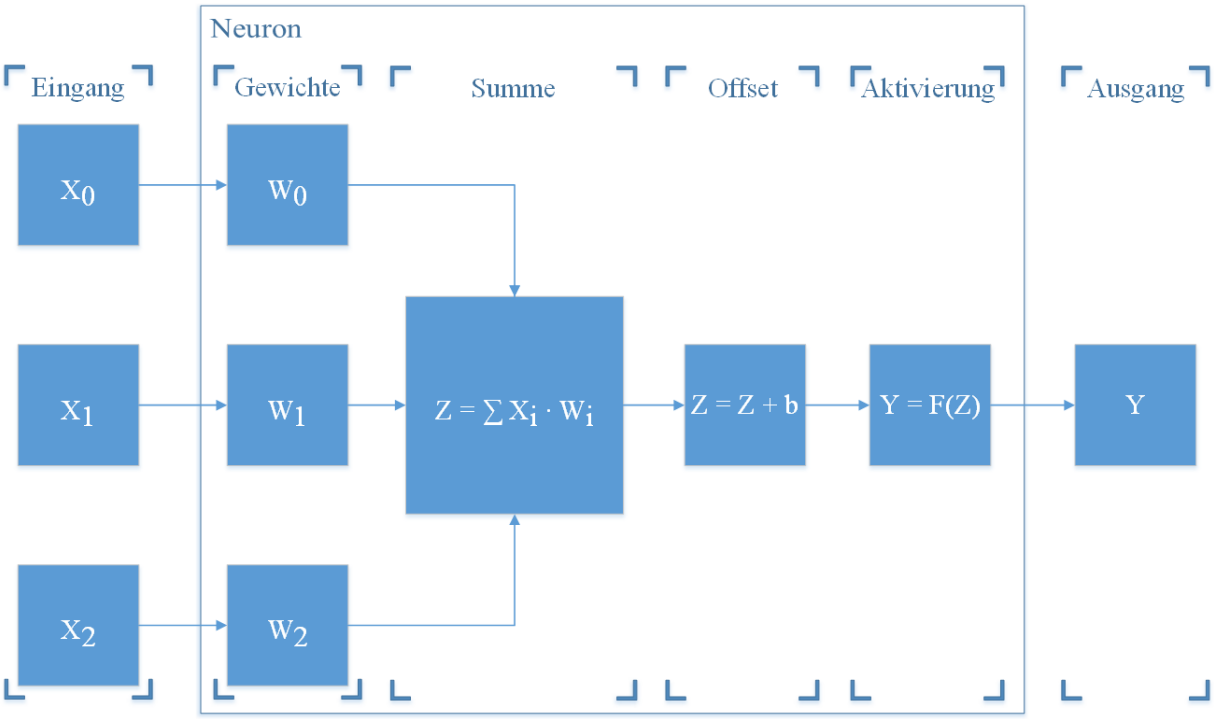

Abbildung 2-10 - Mathematisches Modell eines Neurons

In der Abbildung 2-10 ist das mathematische Modell für ein Neuron dargestellt. Wie bereits beschrieben, wird jeder Eingangsparameter X_i mit einem individuellen Gewicht W_i multipliziert. Anschließend wird die Summe Z der Multiplikation gebildet und ein Offset b dazu addiert. Abschließend wird die Aktivierungsfunktion $F(Z)$ angewendet. Das Ergebnis dieser Funktion ist gleichzeitig der Ausgabewert Y des Neurons. Zusammenfassend kann die Funktion jedes einzelnen Neurons mit der folgenden Formel beschrieben werden.

$$Y = F\left(\left(\sum_{i=0}^{n} X_i \cdot W_i\right) + b\right)$$

Um diese mathematisch modellierten Neuronen in nützlichen Anwendungen einzusetzen, werden meisten viele Schichten mit jeweils einer beliebigen Anzahl an Neuronen eingesetzt.

In der Abbildung 2-11 ist die Struktur eines neuronalen Netzes mit drei versteckten Schichten dargestellt. Alle Schichten zwischen der Eingangs- und der Ausgangsschicht werden als versteckte Schichten bezeichnet. Aus der Abbildung wird deutlich, dass die Schichten eine unterschiedliche Anzahl von Neuronen haben können. Alle Eingangsparameter bzw. alle Neuronen werden mit sämtlichen Neuronen der nächsten Schicht verknüpft. Diese vollständige Verknüpfung ist in vielen Strukturen üblich, aber es gibt auch Strukturen ohne vollständige Verknüpfung. Durch dieses Netz wird aus den drei Eingangsparametern ein Ausgangsvektor berechnet. Die relevantesten derzeit verwendeten Netzwerkstrukturen sind publiziert und

lediglich die Parameter müssen noch auf die Anwendung angepasst werden. Außerdem gibt es noch weitere Arten von Neuronen.

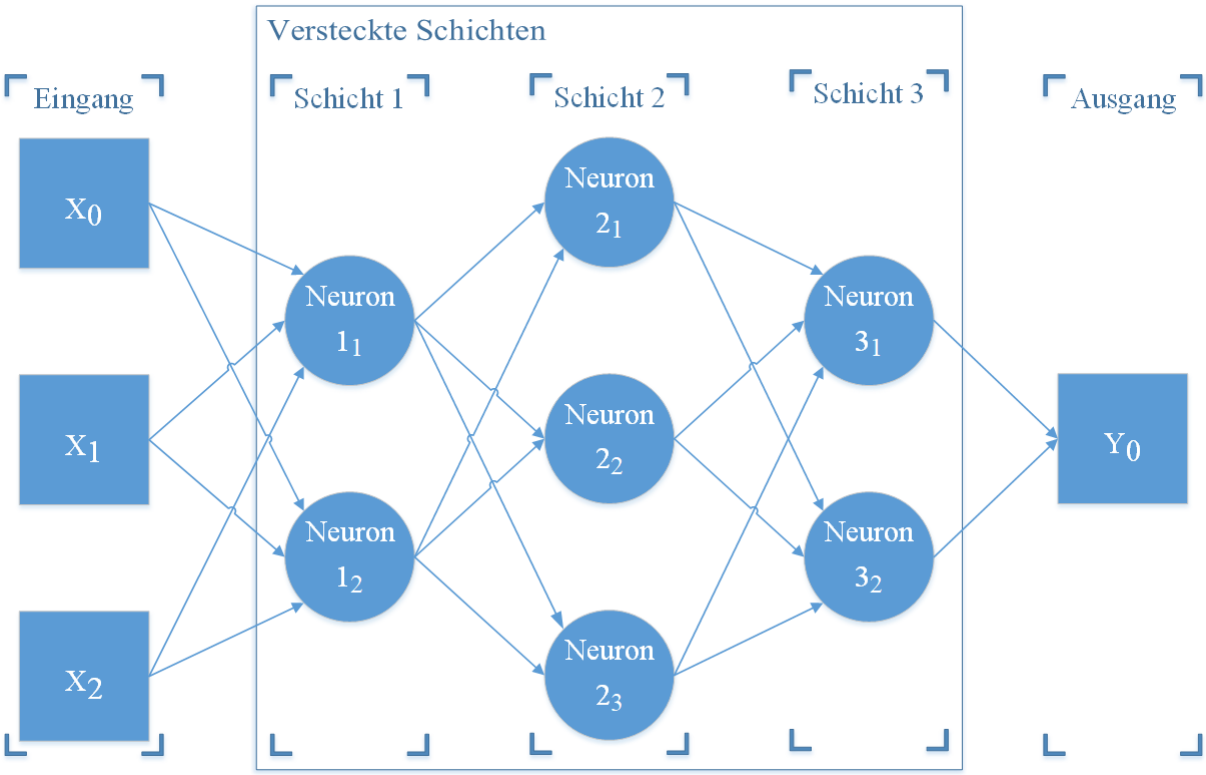

Abbildung 2-11 - Künstliches neuronales Netz mit drei versteckten Schichten

Um neuronale Netze zu trainieren, können alle im Abschnitt 2.3.2 erwähnten Lernverfahren eingesetzt werden. Beim Training ändern die Neuronen ihre Gewichte sowie ihren Offset. Aufgrund der veränderten Matrizen, werden andere Werte an die Aktivierungsfunktion, die damit die Aktivität eines Neurons simuliert, übergeben. Um die Gewichte und den Offset anzupassen, benötigt das neuronale Netz Trainingsdaten.

Beim Lernen mit Lehrer sind die korrekten Ein- und Ausgabevektoren in den Trainingsdaten enthalten und das neuronale Netz kann die Parameter damit einstellen. Durch diese Lernmethode kann beispielsweise eine Erkennung von handschriftlichen Ziffern realisiert werden. Eine Datenbank mit Bildern von handschriftlich Ziffern und die jeweils zugehörigen Werte werden dem neuronalen Netz zum Training übergeben. Anschließend kann es selbstständig handschriftlich geschriebene Ziffern erkennen.

Das Lernen durch Verstärkung funktioniert nicht über vollständig geprüfte Trainingsdaten, sondern über positives oder negatives Feedback. In diesem Fall ist das gewünschte Ergebnis bekannt und es kann bestätigt werden, ob es falsch oder richtig ist. Allerdings kann nicht bestätigt werden, wie richtig oder falsch der Eingabevektor ist. Ein bekanntes Beispiel ist das

Computerprogramm AlphaGo. Dieses Programm hat über verstärkendes Lernen die Spielposition beim Brettspiel Go bewertet und ist zum besten Spieler der Welt geworden.

Beim Lernen ohne Lehrer wird nur der Eingabevektor vorgegeben und die Parameteränderung erfolgt nur in Abhängigkeit der vorgegebenen Eingabevektoren. Der Ausgabevektor ist nicht bekannt. Mit dieser Methode kann die Ähnlichkeit von Eingabedaten festgestellt werden. Daher kann ein Clustering realisiert werden, indem die jeweils ähnlichen Eingabedaten gleich klassifiziert werden. Als Beispiel kann hier die Analyse von Bildern angeführt werden. Es ist nicht bekannt was sich alles auf dem Bild befindet, aber das neuronale Netzwerk kann verschiedene Elemente auf dem Bild wie beispielsweise ein Haus klassifizieren, da es zu bekannten Bildern Ähnlichkeiten herstellen kann.

Für jede Anwendung muss eine eigene Netzstruktur, dass bedeutet die Anzahl der versteckten Schichten mit deren Anzahl an Neuronen, gefunden werden. Durch zu wenige Parameter kann das Netz eventuell nicht generalisieren, aber durch zu viele Parameter lernt das Netz die möglichen Fälle nur auswendig. Der zweite Fall wird als Überanpassung bezeichnet. Dementsprechend muss anwendungsabhängig eine Netzstruktur gefunden werden und durch Überprüfen des Leistungsindexes in Abhängigkeit der Komplexität der Netzstruktur das neuronale Netz bewertet werden. [18]

Kapitel 3

Lernziele für Schülerprojekte und daraus abgeleitete Lerninhalte

Im Rahmen des Kooperationsprojektes letsgoING der Hochschule Reutlingen mit verschiedenen Schulen in der Umgebung sollen diverse Themen aus den Bereichen Datensicherheit und künstliche Intelligenz hinsichtlich ihres Einsatzes im Schulunterricht untersucht werden.

3.1 Datensicherheit

Auf Grund der steigenden Vernetzung nimmt die Anzahl der Geräte, die mit dem Internet verbunden werden, rasant zu. Durch intelligente Haussteuerungen, Smartphones, Tablets und viele weitere Geräte werden große Mengen privater Daten über das Internet gesendet. Das neu entwickelte Projekt zum Internet der Dinge soll den Schülern die Vorteile von kabelloser Vernetzung demonstrieren und die Freude am Erstellen eigener Anwendungen wecken. Diese Arbeit beschäftigt sich mit den Gefahren und Schwachstellen einer zunehmenden Vernetzung und soll die Schüler sensibilisieren, damit sich diese bei allen Vorteilen der Digitalisierung auch den Risiken bewusstwerden. Aufgrund eines unzureichenden Zertifizierungssystems für IT- und Datensicherheit von Produkten wie Apps, Laptops oder Smartphones ist eine Erziehung zu einer digitalen Mündigkeit bereits in der Schule notwendig.

3.1.1 Erläuterung und Analyse des Projekts aus dem Bereich Internet der Dinge

Bei dem Projekt werden über das MQTT Protokoll verschiedene Daten ausgetauscht. Verschiedene Clients sollen Sensordaten auslesen und unter dem jeweiligen Thema an den Broker versenden. Außerdem können die Clients zu verschiedenen Themen Daten empfangen und damit verschiedene Aktoren ansteuern. Mit diesem Projekt kann zum Beispiel eine Hausautomatisierung realisiert werden. Das Netzwerk soll unabhängig vom Internet in einem lokalen Netzwerk funktionieren, damit es in den Schulen eingesetzt werden kann.

Um verschiedene Angriffsszenarien zu entwickeln, ist es wichtig, das System zu analysieren. Beim zentralen Server handelt es sich um einen Raspberry Pi 3 Model B mit dem

Betriebssystem `Raspbian Jessie`, der mit dem Programm `Mosquitto` als MQTT Broker fungiert. Dieser öffnet über das Programm `Hostapd` ein WPA2 verschlüsseltes Netzwerk, in dem die Clients über DHCP ihre IP-Adresse zugewiesen bekommen. Außerdem öffnet er im lokalen Netzwerk eine Website, auf der verschiedene Informationen zur Verwaltung des Netzwerkes und den aktuellen Messungen dargestellt werden können. Als Clients melden sich mehrere WLAN-Module im Netzwerk an. Diese senden und empfangen Informationen über das WLAN.

Die Rückfalllösung des Projektes ist es, dass die WLAN-Module selbst ein Netzwerk öffnen und eine eigene Website hosten. Über diese Website können verschiedene Parameter ausgelesen und diverse Aktoren angesteuert werden. Diese Alternative zum MQTT-Ansatz ist wichtig, da das Projekt möglichst robust im Schuleinsatz funktionieren muss. Fehlt oder fällt der Raspberry Pi aus, können die WLAN-Module ein eigenes Netzwerk öffnen und weiterhin funktionieren, damit die Unterrichtsstunde trotzdem durchgeführt werden kann.

3.1.2 Angriffsmöglichkeiten und Auswahl eines Rechnersystems

Das vorliegende System bietet verschiedene Angriffsmöglichkeiten. Zuerst soll das WPA2 Passwort geknackt oder in Erfahrung gebracht werden. Anschließend soll sich der Angreifer als Mittelsmann zwischen dem WLAN-Modul und dem Raspberry Pi positionieren, um Datenpakete abzuhören und zu manipulieren. Des Weiteren soll ein Angriff auf die Verfügbarkeit der Dienste mithilfe verschiedener Denial-of-Service-Angriffe untersucht werden. Abschließend soll durch eine infizierte Datei die Kontrolle über das Betriebssystem erlangt werden.

Die Lerneinheiten zum Thema Datensicherheit werden in den Schulen vorgeführt, daher ist eine preiswerte und mobile Hardware sowie ein robustes Betriebssystem mit bereits vorinstallierter Software notwendig. Aus diesem Grund werden alle Angriffe ebenfalls mit einem Raspberry Pi 3 Model B realisiert und getestet. Der Einplatinencomputer liefert genug Rechenleistung für die Angriffe bzw. deren Simulation. Außerdem ist er preiswert und gleichzeitig sehr mobil. Als Betriebssystem wird das von der Firma Offensive Security entwickelte `Kali Linux` eingesetzt, da es zahlreiche Programme für Sicherheitstest bereits vorinstalliert hat und ein Image für den Raspberry Pi bereitgestellt wird. Der einzige Nachteil an diesem Betriebssystem ist, dass erste Linux Kenntnisse für die Angriffe vorhanden sein sollten. Allerdings verfügen viele Programme über eine grafische Benutzeroberfläche. Aus diesem Grund wird der Einstieg in das Betriebssystem erleichtert.

3.2 Künstliche Intelligenz

Im Zuge der Digitalisierung wird das tägliche Leben zunehmend durch künstliche Intelligenzen wie Sprachassistenten, selbstfahrende Autos oder maschinelle Datenanalysen beeinflusst. Die meisten funktionieren nur durch Analyse und auch Speicherung von Benutzerdaten. Aus diesem Grund wird es immer wichtiger, sich mit dem Technologietrend zu beschäftigen und zu entscheiden, in welchem Werteverhältnis die Veröffentlichung privater Daten zu dem jeweiligen Nutzen steht.

Im Folgenden sollen drei mögliche Anwendungsszenarien künstlicher Intelligenz vorgestellt werden. Um eine plattformunabhängige Lösung zu realisieren, sollen die Szenarien alle mit dem Open-Source Programm `TensorFlow` in der Programmiersprache Python erstellt werden können.

3.2.1 Anwendungsszenario A: Autonomer Linienfolger

Im Kooperationsprojekt wird bereits seit mehreren Jahren erfolgreich der ArduRover eingesetzt. Es handelt sich um ein Fahrzeug mit zwei Gleichstrommotoren an der Vorderachse, die über Pulsweitenmodulation angesteuert werden können sowie einer Rolle an der Hinterachse zur Stabilisierung. Außerdem ist das Fahrzeug mit diversen Sensoren ausgestattet, damit es einer schwarzen Linie auf weißem Untergrund folgen kann. Diese Funktion wird in kleinen Arbeitspaketen von den Schülern unter der Betreuung von Studenten erarbeitet. Die Erkennung der Linie erfolgt über eine Infrarot-LED und jeweils einem Lichttransistor auf jeder Seite der LED. Dieser Aufbau wird auch als differentieller Lichttaster bezeichnet. Innerhalb des Projektes letsgoING wird die Sensoranordnung als Linienfolgesensor erläutert. Die beiden Messergebnisse der Transistoren werden von einem Arduino Uno ausgelesen. Anschließend wird die Differenz der beiden Werte gebildet, damit die Steuerung unabhängig von äußeren Störfaktoren wie zum Beispiel Sonnenlicht ist. Nun wird ab einer bestimmten Differenz eine Rechtskurve oder eine Linkskurve gefahren. Diese Regelung erfolgt mittels einer klassischen Zweipunktreglung. Zum Fahren der jeweiligen Kurve wird der kurveninnere Motor mit einer kleineren Modulationsfrequenz als der kurvenäußere Motor angesteuert.

Die beschriebene Zweipunktregelung soll durch ein neuronales Netz ersetzt werden. Als Eingangsparameter stehen hierbei die beiden Werte der Lichttaster zur Verfügung und die Ausgangsparameter sind die Ansteuerungswerte für die Motoren. Durch Trainingsdaten mit bekannten Eingabe- und Ausgabevektoren wird die gewünschte Funktion antrainiert und durch Testdaten mit ebenfalls bekannten Vektoren verifiziert. Das Training des neuronalen Netzes wird auf einem leistungsstärkeren Rechner durchgeführt. Abschließend soll das Leistungsmaß

bewertet und überprüft werden, ob die Ausführung des neuronalen Netzes auf dem Arduino Uno möglich ist.

3.2.2 Anwendungsszenario B: Klassifizierung verschiedener Körpern

Die zweite Idee stützt sich auf einen Artikel in der Zeitschrift c't. Die Zeitschrift behandelt sämtliche Themen im Bereich Informatik und erstellt regelmäßig Tutorials für Anwender, um Themen selbstständig anwenden und vertiefen zu können. Der Artikel beschreibt die Anwendung eines neuronalen Netzes zur Klassifizierung drei verschiedener Körper. Hierzu rastert ein Infrarotsensor innerhalb einer Sekunde einen Körper ab und nimmt dabei 24 Messwert auf. Dadurch entsteht ein Profilbild der Oberfläche, die dem Infrarotsensor zugewandt ist. Dieses wird über ein neuronales Netz mit einer versteckten Schicht, die aus sechs Neuronen besteht, ausgewertet. Am Ausgang werden drei unstetige Parameter ausgegeben. Damit die Eingangsparameter gleich sind, werden die Körper während der Messung auf einem Förderband an dem Infrarotsensor vorbeigeführt. Neben einer detaillierten Anwendungsbeschreibung werden die zugehörigen Trainings- und Testdaten sowie GitHub Repositorien zur Verfügung gestellt. [19]

3.2.3 Anwendungsszenario C: Klassifizierung von Zuständen im Klassenraum

Die dritte Anwendung baut auf dem im Abschnitt 3.1.1 vorgestellten Projekt zum Internet der Dinge auf. Hier könnte der zentrale MQTT-Broker durch eine Datenanalyse verschiedene Zustände im Klassenraum unterscheiden. Die räumlichen verteilten WLAN-Module würden verschiedene Sensordaten wie zum Beispiel Luftdruck, Luftfeuchtigkeit, Gaskonzentration und Temperatur an den Broker übermitteln. Dieser könnte mithilfe eines neuronalen Netzes die Daten analysieren und verschiedene Zustände im Klassenraum wie beispielsweise Unterricht, Nachtruhe oder Klausur klassifizieren. In Abhängigkeit der Genauigkeit der Daten wäre es sogar denkbar, dass die Anzahl der Personen in einem Klassenraum vorhergesagt werden kann. Unter Umständen könnte hier ein Lernverfahren ohne Lehrer eingesetzt werden, welches über Clustering verschiedene Zustände im Klassenraum klassifiziert. Des Weiteren könnte aufgezeigt werden, dass über die Fusion mehrere unbedenklicher Sensordaten sensible Informationen wie beispielsweise die Anzahl der Personen in einem Raum generiert werden kann.

Kapitel 4

Schülerprojekte im Bereich Datensicherheit

Die Untersuchungen aus dem Abschnitt 3.1 sollen in diesem Kapitel hinsichtlich ihrer Umsetzung analysiert werden und ein Ansatz zur Realisierung entwickelt werden. Anschließend sollen die Projekte aus dem Bereich Datensicherheit auf ihrer Funktionalität getestet werden.

4.1 Übersicht

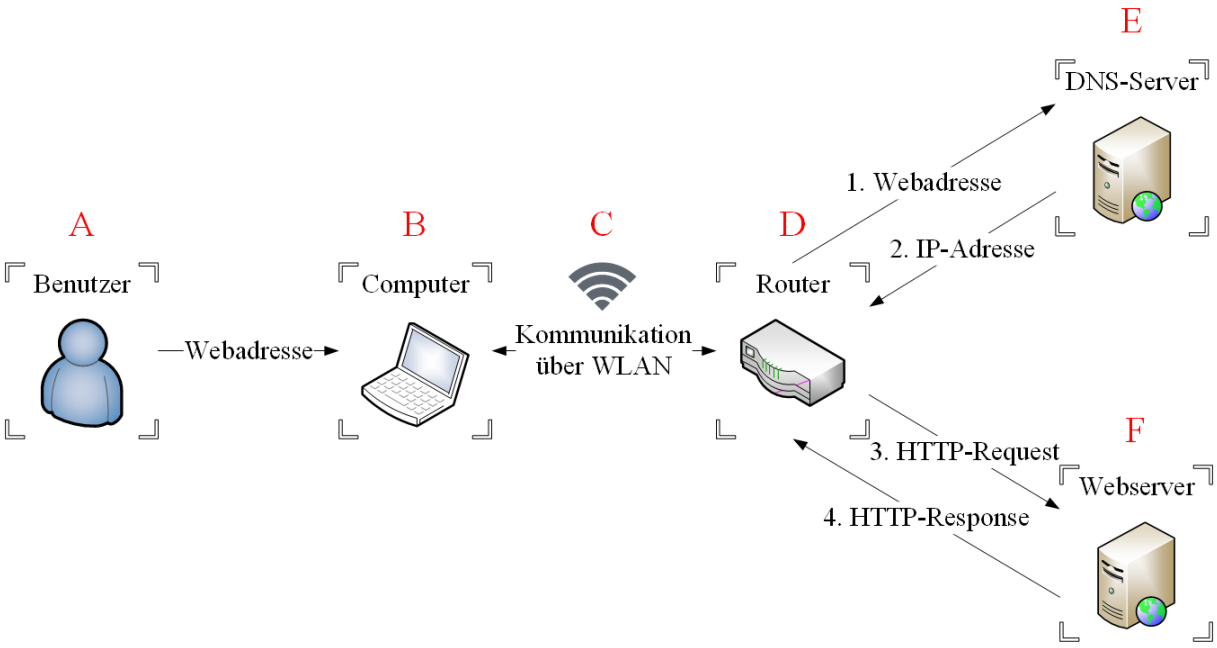

Abbildung 4-1 - Aufruf einer Webadresse im Internet mit Markierung der Schwachstellen

In der Abbildung 4-1 ist der Aufruf einer Webadresse erneut dargestellt. Allerdings wurden die einzelnen Komponenten des Systems markiert. In den folgenden Abschnitten werden verschiedene Angriffe auf die unterschiedlichen Komponenten durchgeführt. Die angegriffene Komponente wird anhand dieser Abbildung verdeutlicht. Dadurch kann die verwendete Schwachstelle besser nachvollzogen werden.

4.2 WPA2-Wörterbuch-Attacke

Die Wörterbuch-Attacke wird auch als Brute-Force-Angriff bezeichnet. Dieser Ausdruck bedeutet brachiale Gewalt und beschreibt den Vorgang treffend, denn hierbei wird versucht durch stupides ausprobieren beliebig vieler Passwörter in das WPA2 verschlüsselte Netzwerk einzudringen. Dieses Vorgehen ist der Tatsache geschuldet, dass die WPA2 Verschlüsselung noch nicht entschlüsselt ist. Die früher verwendete WEP Verschlüsselung gilt als entschlüsselt. Bei dieser unsicheren Verschlüsselung ist kein Erraten des Passwortes nötig, denn es kann durch mitgeschnittenen Netzwerkverkehr rekonstruiert bzw. berechnet werden.

In der Abbildung 4-1 wird die Schwachstelle C angegriffen. Die Kommunikation über WLAN soll entschlüsselt werden.

4.2.1 Durchführung der WPA2-Wörterbuch-Attacke

Um den Angriff vorzubereiten, muss der WLAN Chip in den Monitormodus versetzt werden. Hierfür sowie für den weiteren Angriff kann das Programm `Aircrack` verwendet werden. Das kostenlose Programm verfügt einen breiten Funktionsumfang, um drahtlose Netzwerke zu analysieren und zu attackieren.

Der Monitormode ermöglicht es sämtliche verfügbaren Netzwerke zu scannen und für ein ausgewähltes Netzwerk den Handshake zwischen dem Router und einem Client mitzuschneiden sowie abzuspeichern. Entweder wird eine Anmeldung eines Clients abgewartet oder mittels einer Deauthentication-Attacke eine erneute Anmeldung erzwungen. Sobald ein Handshake aufgezeichnet wurde, besteht die Möglichkeit Passwörter aus einem Wörterbuch auszuprobieren. Die Passwörter werden nicht direkt am Router ausprobiert, sondern im Handshake abgeglichen. Daher kann der Router den Angriff nicht erkennen und den Angreifer blockieren. Außerdem können deutlich mehr Passwörter pro Sekunde ausprobiert werden, da keine Kommunikation über WLAN stattfinden muss. Sobald ein Passwort mit den mitgeschnittenen Nonce-Werten und den MAC-Adressen den Pairwise Transient Key PTK ergibt, ist das korrekte Passwort gefunden.

Das Wörterbuch wird selbst erstellt, aus dem Internet heruntergeladen oder mit vollautomatisierten Programmen nach Belieben generiert.

Zur Validierung wird als Angriffsrechner ebenfalls der Raspberry Pi 3 Model B verwendet. Mit dem erwähnten Programm `Aircrack` sowie der beschriebenen Vorgehensweise, kann das WPA2 Netzwerk des MQTT-Servers angegriffen werden.

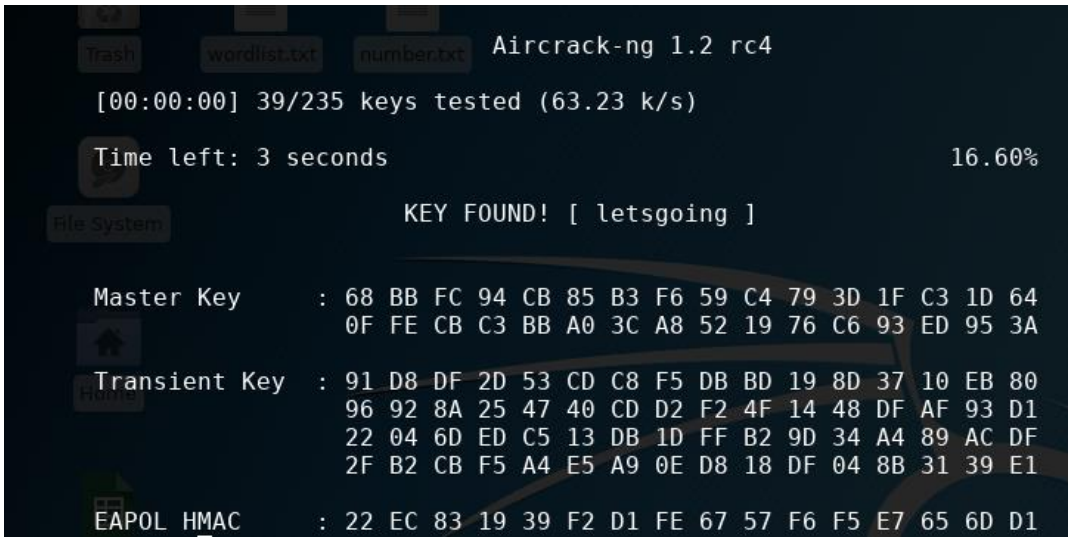

Abbildung 4-2 - Wörterbuchattacke auf ein WPA2 Netzwerk

In der Abbildung 4-2 ist der erfolgreiche Abschluss des Angriffs dargestellt. Der Angreifer hat ein Wörterbuch mit 235 Passwörtern eingesetzt und ist dabei nach 39 Passwörtern auf das richtige Passwort „letsgoing“ gestoßen. Nach 16,6% des Wörterbuches wurde demnach das Passwort herausgefunden. Die Geschwindigkeit des Angriffs hängt dabei stark von der Rechenleistung des Angriffsrechners ab. Das Programm `Aircrack` gibt diese Geschwindigkeit in Passwörtern pro Sekunde an. Das bedeutet, wie viele Passwörter pro Sekunde ausprobiert werden, um das Passwort des Zielnetzwerkes zu entschlüsseln. Bei dem oben genannten Rechner liegt die Geschwindigkeit bei ungefähr 110 Passwörtern pro Sekunde. Das entspricht einer Geschwindigkeit von 6600 Passwörtern pro Minute.

4.2.2 Lerninhalte und Analyse der Rechenzeit

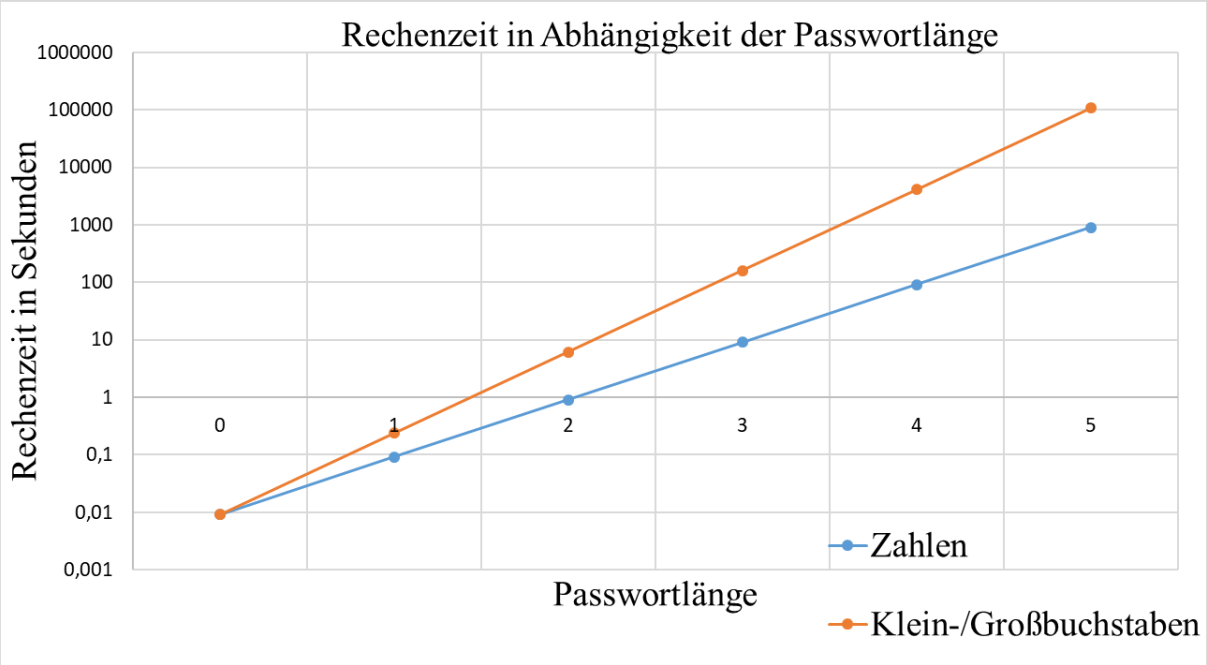

Abbildung 4-3 - Abhängigkeit zwischen Rechenzeit, Passwortlänge und Kombinationen

In der Abbildung 4-3 ist der Zusammenhang zwischen Rechenzeit, Passwortlänge und möglichen Kombinationen dargestellt. Die Rechenzeit bezieht sich auf die Geschwindigkeit von 110 Passwörtern pro Sekunde. Die blaue Linie stellt die Rechenzeit für Kombinationen aus den Zahlen null bis neun dar. Die orange Linie stellt die Rechenzeit für Kombinationen aus entweder Klein- oder Großbuchstaben dar. Die steilere Steigung der orangenen Kennlinie hängt mit der größeren Anzahl an Kombinationsmöglichkeiten zusammen.

Anhand dieser Rechenzeiten ist es im Rahmen des Projektes letsgoING sinnvoll, den Schülern die Berechnung von Passwörtern aus vierstelligen Zahlenkombinationen oder dreistelligen Buchstabenkombinationen zu demonstrieren. Die Zahlenkombinationen können im schlechtesten Fall innerhalb von knapp 90 Sekunden und die Buchstabenkombinationen im schlechtesten Fall innerhalb von 160 Sekunden auf dem Raspberry Pi 3 Model B berechnet werden. Diese kurze Zeit eignet sich, um im Rahmen des Projektes die Angriffsmethode zu erläutern und anhand eines praktischen Beispiels die Anwendung zu verdeutlichen.

Außerdem werden die Schüler für eine gewissenhafte Auswahl von Passwörtern sensibilisiert, indem beispielsweise aufgezeigt wird, wie unsicher Passwörter aus nur Zahlenkombinationen oder Standardpasswörter sind. Auf der anderen Seite kann gezeigt werden, wie einfach es ist, ein sehr sicheres Passwort zu erstellen. Neben einer ausreichenden Passwortlänge sollten Zahlen, Klein- und Großbuchstaben sowie Sonderzeichen kombiniert werden. Werden alle diese Elemente in einem Passwort verwendet, so erhöht sich die Rechenzeit deutlich, denn es

stehen 86 verschiedene Zeichen zur Verfügung. Im Rahmen von Passwörtern für das eigene Heimnetzwerk sollte darauf hingewiesen werden, dass der WEP Standard nicht mehr sicher ist und daher im Idealfall eine WPA2-Verschlüsslung oder mindestens eine WPA-Verschlüsselung für das Heimnetzwerk verwendet werden sollte.

Zu beachten ist, dass die Rechengeschwindigkeit auf dem Raspberry Pi 3 Model B als gering einzustufen ist. Leistungsstarke Computer können Rechengeschwindigkeiten von mehreren Milliarden Passwörtern pro Sekunde erreichen. Mit solchen enormen Rechenleistungen lassen sich auch längere Passwörter in kurzer Zeit errechnen. Außerdem ist zu beachten, dass es kein Problem darstellt, einen Rechner mehrere Tage oder Wochen rechnen zu lassen. Der in Abbildung 4-3 rechenaufwendigste dargestellte Fall für ein fünfstelliges Passwort aus nur Klein- oder nur Großbuchstaben ist mit der Rechengeschwindigkeit von nur 110 Passwörtern pro Sekunde ebenfalls in maximal 30 Stunden errechnet. Vor einer Wörterbuchattacke gibt es keinen effektiven Schutz, außer der Verwendung aktueller Sicherheitsstandards und Passwörter, die einen ausreichenden Schutz bieten.

In einer Auswertung von 734000 Passwörtern waren über 30% der Passwörter zwischen einem und sechs Zeichen lang. Damit sind diese alle in angemessener Rechenzeit berechenbar. Des Weiteren verwendeten 17000 Personen das Passwort „123456“ und fast 4600 Personen verwendeten das Passwort „password“. Diese Statistik zeigt auf, dass Wörterbuch-Attacken, die unter anderem auch Standardpasswörter wie „123456“ oder „password“ enthalten, eine erstaunlich hohe Erfolgsquote haben. [20]

4.3 WPA2-Phishing-Attacke

Phishing ist eine weitere Methode ein unbekanntes Passwort zu erhalten. Die aus dem Social Engineering bekannte Methode wird hierbei eingesetzt, um den Benutzer dazu zu bewegen, sein Passwort freiwillig weiterzugeben.

In der Abbildung 4-1 wird die Schwachstelle A attackiert. Der Benutzer verrät hierbei freiwillig das Passwort, welches zur Verschlüsselung der Kommunikation über WLAN verwendet wird, an den Angreifer.

4.3.1 Durchführung der WPA2-Phishing-Attacke

Zuerst werden die verfügbaren Netzwerke im Monitormodus gescannt und der Handshake vom ausgewählten Netzwerk gespeichert. Anschließend kopiert der Angreifer das Zielnetzwerk und öffnet einen Zwilling mit den gleichen Parametern wie SSID, MAC-Adresse und Kanal. Daraufhin können die mit dem echten Netzwerk verbunden Clients mit einer Deauthentication-

Attacke zu einer erneuten Anmeldung gezwungen werden oder der Angreifer wartet bis sich ein Teilnehmer in seinem Netzwerk anmeldet. Versucht ein Client sich in das Netzwerk des Angreifers anzumelden, nutzt dieser das Passwort sowie den vorher aufgezeichneten Handshake, um das Passwort zu verifizieren. In der Regel benutzt ein Client das Netzwerk mit dem stärksten Empfang. Sollte das Passwort nicht stimmen, wird der Client zur erneuten Eingabe des Passworts aufgefordert. Wenn der Client das richtige Passwort herausgegeben hat, wird dieses gespeichert und der Client an das richtige Netzwerk weitergeleitet. Sobald das richtige Passwort gefunden ist, wird das vom Angreifer erstellte Netzwerk geschlossen. In der Abbildung 4-4 wurde der oben beschrieben Ablauf unter Vernachlässigung einer möglichen Deauthentication-Attacke dargestellt.

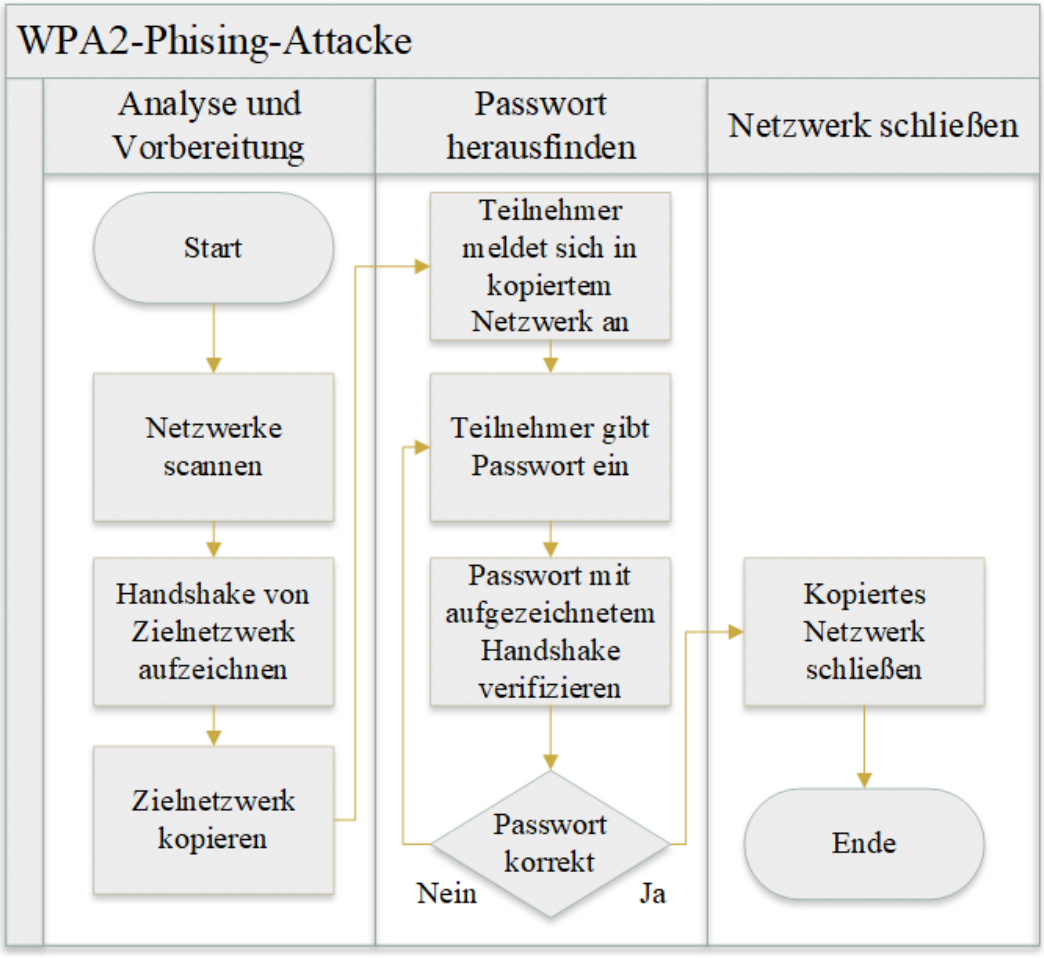

Abbildung 4-4 - Ablaufdiagramm der WPA2-Phishing-Attacke

Durch diesen Angriff soll das Passwort durch den Benutzer unbewusst an den Angreifer verraten werden. Der Angreifer kopiert ein Netzwerk und sobald sich jemand bei dem kopierten Netzwerk anmeldet, fordert er den neuen Teilnehmer auf, das Passwort einzugeben, um Internetzugang zu erhalten. Es können verschiedene Anmeldebildschirme verwendet werden.

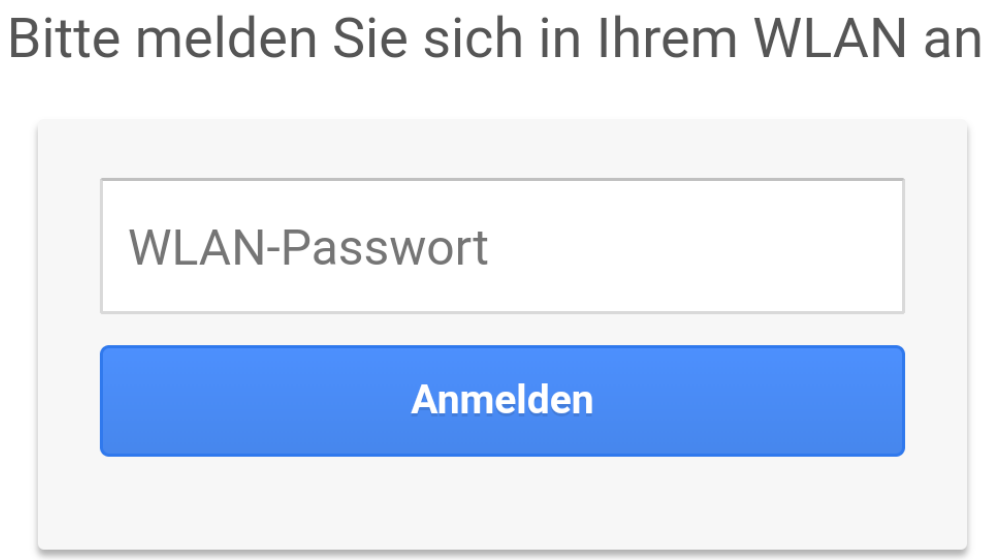

Abbildung 4-5 - Interface des gefälschten Netzwerkes

In der Abbildung 4-5 ist ein solcher Anmeldbildschirm dargestellt. Dieser fordert den Teilnehmer auf, dass er das WLAN-Passwort nochmals eingibt. Sobald er dieses Passwort eingegeben hat, verifiziert der Angreifer das Passwort am aufgezeichneten Handshake.

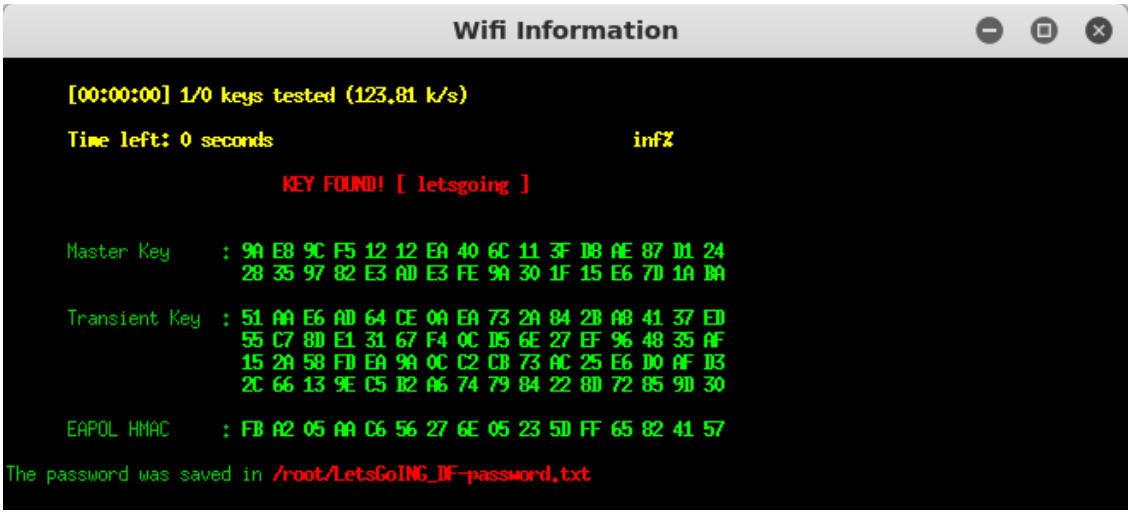

Abbildung 4-6 - Erfolgreicher Phishing-Angriff auf ein Netzwerk

Wenn das eingebende Passwort korrekt ist, wird das gefälschte Netzwerk geschlossen und das Opfer an das korrekte Netzwerk weitergeleitet. Die Abbildung 4-6 stellt den Abschluss des erfolgreichen Angriffs dar. Das Netzwerk wurde geschlossen und das Passwort verifiziert. Das korrekte Passwort „letsgoing“ wurde ebenfalls gefunden.

4.3.2 Lerninhalte und Realisierungsprobleme

Durch diesen Angriff kann aufgezeigt werden, wie leicht die Schüler selbst Opfer einer Phishing-Attacke werden. Vor allem im Bereich öffentlicher Netzwerke kann hiermit sehr leicht das Passwort erspäht werden. Die Anmeldebildschirme sind in diesem Bereich üblich.

Im gewählten Setup aus dem Raspberry Pi und dem WLAN-Modul wird dieser Angriff allerdings nicht funktionieren, da das WLAN-Modul den Anmeldebildschirm nicht verwenden kann. Allerdings könnte dieser Angriff verwendet werden, wenn das WLAN-Modul als Webserver fungiert und die Schüler sich mit ihren Smartphones anmelden. Dann könnte das WLAN kopiert werden und die Schüler das Passwort beim Einloggen verraten.

Ein weiteres Problem ist es, dass der Angriff nicht auf dem Raspberry Pi ausgeführt werden konnte. Dieser müsste hierzu einen USB-WLAN-Dongles erkennen, da die verwendete Software den internen WLAN-Chip des Raspberry Pi nicht erkennt. Mit dem verwendeten Betriebssystem `Kali Linux` war es nicht möglich den WLAN-Dongle auf dem Raspberry Pi zu verwenden.

4.4 Man-in-the-Middle

Die in den Abschnitten 4.2 und 4.3 vorgestellten Angriffe sind notwendig, um die Grundlage für eine Man-in-the-Middle-Attacke zu legen. Nur wenn der Angreifer das Passwort eines Netzwerkes kennt, kann er den Datenverkehr entschlüsseln. Ein Ende-zu-Ende verschlüsselter Datenverkehr ist für den Angreifer nicht lesbar und daher unrelevant.

In der Abbildung 4-1 werden die Schwachstelle B und D durch ARP-Spoofing getäuscht. Sowohl der Router als auch der Computer verwenden das ARP-Protokoll, um im WLAN zu kommunizieren. Diese sind meistens unzureichend gegen das ARP-Spoofing geschützt. In den Abschnitten 2.1.2, 2.2.3 und 2.2.4 werden die notwendigen Grundlagen für den Angriff erläutert.

4.4.1 Manipulation der ARP-Cache

Das ARP-Spoofing kann mit dem Programm `Ettercap` durchgeführt werden, indem auch verschiedene Filter zur Datenmanipulation erstellt werden können.

Wie bereits erläutert, werden die im Netzwerk vorhandenen Teilnehmer per ARP-Spoofing dazu gezwungen, ihre Daten direkt über den Angreifer zu senden. Bei den Netzwerkteilnehmern handelt es sich um den Raspberry Pi 3 Model B als MQTT-Broker sowie das WLAN-Modul als MQTT-Client.

```
pi@LETSGOING:~ $ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
espressif                ether   30:ae:a4:05:f4:00   C                     wlan0
kali                     ether   b8:27:eb:f8:ad:96   C                     wlan0
rtec-me.fh-reutlingen.d  ether   2c:5a:0f:25:b3:00   C                     eth0
```

Abbildung 4-7 - ARP-Cache des MQTT-Brokers vor ARP-Spoofing

Die Abbildung 4-7 stellt die ARP-Cache des MQTT-Brokers dar. Neben dem WLAN-Modul „espressif“ mit der MAC-Adresse 30:ae:a4:05:f4:00 befindet sich ebenfalls der Angreifer „kali“ mit der MAC-Adresse b8:27:eb:f8:ad:96 im Netzwerk. Zu diesem Zeitpunkt sind die Hardwareadressen den korrekten Netzwerkteilnehmern zugeordnet.

```
pi@LETSGOING:~ $ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
espressif                ether   b8:27:eb:f8:ad:96   C                     wlan0
kali                     ether   b8:27:eb:f8:ad:96   C                     wlan0
rtec-me.fh-reutlingen.d  ether   2c:5a:0f:25:b3:00   C                     eth0
```

Abbildung 4-8 - ARP-Cache des MQTT-Brokers nach ARP-Spoofing

Nachdem das ARP-Spoofing erfolgreich durchgeführt wurde, ändert sich, wie in der Abbildung 4-8 dargestellt, die ARP-Cache des MQTT-Brokers. Dieser kennt noch immer die gleichen Teilnehmer „espressif“ und „kali“, allerdings ordnet er jetzt den beiden Teilnehmern die MAC-Adresse des Angreifers zu. Die manipulierte ARP-Cache validiert die Man-in-the-Middle-Position des Angreifers.

```
▸ Frame 7: 91 bytes on wire (728 bits), 91 bytes captured (728 bits) on interface 0
▸ Ethernet II, Src: Espressi_05:f4:00 (30:ae:a4:05:f4:00), Dst: Raspberr_f8:ad:96 (b8:27:eb:f8:ad:96)
▸ Internet Protocol Version 4, Src: 192.168.1.16, Dst: 192.168.1.1
▸ Transmission Control Protocol, Src Port: 13072, Dst Port: 1883, Seq: 1, Ack: 1, Len: 37
▸ MQ Telemetry Transport Protocol, Publish Message

0000  b8 27 eb f8 ad 96 30 ae  a4 05 f4 00 08 00 45 00   .'....0. ......E.
0010  00 4d 00 3c 00 00 ff 06  38 0d c0 a8 01 10 c0 a8   .M.<.... 8.......
0020  01 01 33 10 07 5b 00 00  1f 95 90 bc a0 30 50 18   ..3..[.. .....0P.
0030  16 5b 3a 5d 00 00 30 23  00 08 6f 75 74 54 6f 70   .[:]..0# ..outTop
0040  69 63 48 65 6c 6c 6f 20  4d 6f 73 71 75 69 74 74   icHello  Mosquitt
0050  6f 20 2d 20 45 53 50 33  32 5f 31                  o - ESP3 2 1
```

Abbildung 4-9 - Analyse des Datenverkehrs im MQTT-Netzwerk durch den Angreifer

In der Abbildung 4-9 ist ein Datenpaket im Netzwerk dargestellt. Nachdem das ARP-Spoofing erfolgreich durchgeführt wurde, können mit Analyseprogrammen wie beispielsweise `Wireshark` die gesendeten Daten der Opfer analysiert werden. Das in der Abbildung 4-9 dargestellte Datenpaket beinhaltet zahlreiche Informationen. Sowohl der Absender „Espressi“ als auch der Empfänger „Raspberr“ werden mit ihrer MAC-Adresse dargestellt. Diese erste Erkenntnis gibt Aufschluss über die verwendete Hardware. Außerdem handelt es sich um eine MQTT-Publish-Nachricht, die auf dem TCP-Port 1883 empfangen wird. Abschließend kann der Angreifer sogar das Thema und die Nachricht „outTopicHello Mosquitto - ESP32_1“ mitlesen. Allerdings kann hierbei nicht zwischen Thema und Nachricht unterschieden werden,

da es zusammenhängend übertragen wird. Sonderzeichen wie der Unterstrich in „ES32_1“ können in `Wireshark` nicht im Klartext gelesen, sondern nur über die hexadezimale Darstellung herausgefunden werden.

4.4.2 Manipulation von Datenpaketen

Mit den aus `Wireshark` gewonnen Kenntnissen sollen die Nachrichten nicht nur mitgeschnitten, sondern auch manipuliert werden. Hierzu bietet `Ettercap` die Möglichkeit, verschiedene Filter zu erstellen und den Datenverkehr somit gezielt zu verändern. Statt der vom WLAN-Modul gesendeten Nachricht „Hello Mosquitto - ESP32_1“ soll die manipulierte Nachricht „Hello Mosquitto -Attacker“ den MQTT-Broker erreichen.

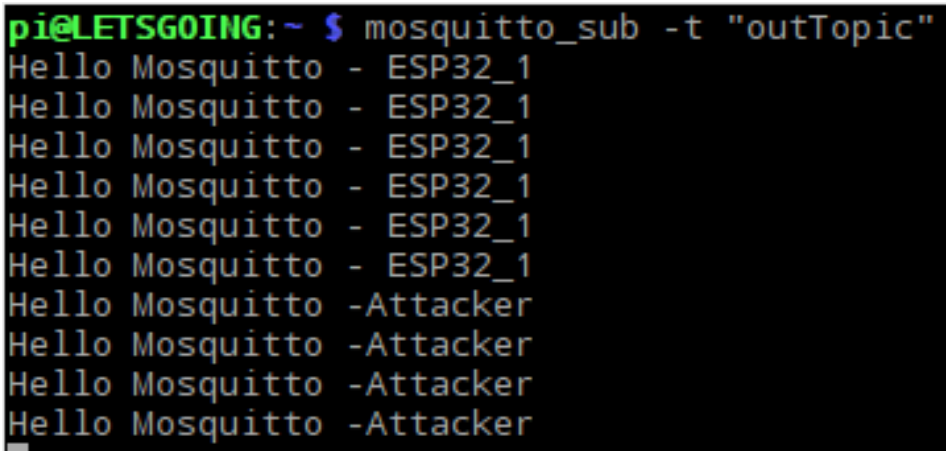

Abbildung 4-10 - Manipulierte MQTT-Nachrichten

Für die beschriebene Manipulation wurde ein Filter erstellt und in der Abbildung 4-10 wurde der Effekt dargestellt. Zuerst abonniert der MQTT-Broker selbst das Thema „outTopic“. Daraufhin empfängt er die vom WLAN-Modul gesendete Nachrichte „Hello Mosquitto - ESP32_1“. Nach sechs erfolgreich und korrekt gesendeten Nachrichten, aktiviert der Angreifer den Filter. Daraufhin empfängt der MQTT-Broker die bereits spezifizierte Nachricht „Hello Mosquitto -Attacker“. Allerdings hat diese Manipulation eine Einschränkung, denn wenn nur die Nachricht manipuliert wird, muss die vom WLAN-Modul verwendete Nachrichtenlänge eingehalten werden, da diese über den TCP Header vom Empfänger überprüft wird. Soll die Nachricht so manipuliert werden, dass sie länger oder kürzer als die ursprüngliche Nachricht ist, muss ebenfalls die mitgesendete Datenlänge angepasst werden.

4.4.3 Lerninhalte und mögliche Abwehrmaßnahmen

Mit diesem Angriff kann den Schülern aufgezeigt werden, wie wichtig es ist, eigene Passwörter geheim zu halten und im Falle des heimischen WLAN Netzwerkes dieses nur an vertraute Personen weiterzugeben oder ein Gästenetzwerk einzurichten. Des Weiteren wird aufgezeigt,

dass unverschlüsselter Datenverkehr ein Sicherheitsrisiko darstellt. Dies ist im Besonderen zu beachten, wenn ein öffentliches Netzwerk benutzt wird. Dritte können unverschlüsselte Passwörter ausspähen, private Daten erfahren oder auch Datenpakete verändern. Aus diesem Grund sollten in öffentlichen Netzwerken niemals Passwörter oder persönliche Daten unverschlüsselt übertragen werden. Bei einer sicherheitskritischen Anwendung, wie zum Beispiel Onlinebanking, ist auf eine verschlüsselte Übertragung mit dem HTTPS-Standard zu achten.

Um das ARP-Spoofing zu erkennen gibt es verschieden Ansätze. Der einfachste Ansatz ist die statische Zuordnung zwischen IP- und MAC-Adressen. Allerdings ist diese Möglichkeit sehr ineffizient, da die ARP-Cache häufig aktualisiert werden muss. Ein weiterer Ansatz ist das Überprüfen der MAC-Adressen in der ARP-Cache, denn normalerweise kommt eine MAC-Adresse nicht zweimal vor. Problematisch an diesem Ansatz ist allerdings die endgültige Identifizierung des Angreifers, da die IP-Adressen oft nicht den Netzwerkteilnehmern zugeordneten werden können und somit der Angreifer nicht vollautomatisiert entfernt werden kann. Der sinnvollste Ansatz ist das strenge Überwachen der ARP-Cache, denn im vorgestellten Angriff können ohne vorherige ARP-Anfragen auch ARP-Antworten gesendet werden. Eine intelligente Überwachung des ARP-Verkehrs, die Zusammenhänge zwischen ARP-Anfragen und ARP-Antworten erstellt und somit manipulierte ARP-Antworten erkennt, löst das Problem des ARP-Spoofing am besten. Hiermit können nicht alle derartigen Attacken erkannt werden, aber der größte Anteil herausgefiltert werden. Das Überwachen des ARP-Verkehrs ist allerdings Aufgabe des Betriebssystems und demnach wird es für leichtgewichtige Mikrocontroller schwer eine notwendige Überwachung zu realisieren.

4.5 Denial of Service

Zur Demonstration von einem Denial-of-Service-Angriff werden drei verschiedene Angriffsmöglichkeiten erläutert, die in Abhängigkeit von der endgültigen Systemkonfiguration verwendet werden können. Alle Angriffe zielen auf die Blockierung eines Dienstes ab und werden als nicht verteilte Angriffe auf nur einem Rechner ausgeführt. Anschließend soll eine Angriffsmöglichkeit getestet werden und mögliche Abwehrmaßnahmen untersucht werden.

4.5.1 Angriffsmöglichkeit A: Der Webserver

In der Abbildung 4-1 wird die Schwachstelle F angegriffen. In diesem Fall soll die Verfügbarkeit des Webservers eingeschränkt werden.

Der erste Angriff zielt auf die Verfügbarkeit der Website des Raspberry Pi oder des WLAN-Moduls ab. Webserver können nur eine begrenzte Anzahl an Verbindungen gleichzeitig halten. Ein einzelner Rechner kann bereits genügend Verbindungen zum Zielserver aufbauen und diese halten, um den jeweiligen Webserver für reguläre Anfrage unerreichbar zu machen. Nachdem die Verbindungen geöffnet wurden, werden regelmäßig weitere Teilanfragen gesendet. Die Anfragen werden nie komplett abgeschlossen, dadurch hält der Webserver die Verbindungen offen, da er auf die Vervollständigung der Anfragen wartet. Durch diese Teilanfragen werden die Verbindungen nicht geschlossen und der Webserver überlastet. Wie in der Abbildung 4-11 visualisiert, ist der Webserver für einen regulären Client nicht mehr erreichbar. Er sendet eine vollständige HTTP-Anfrage an den Webserver, erhält aber keine Antwort mehr auf die Anfrage, da der Webserver mit den Teilanfragen des Angreifers überlastet ist. [14]

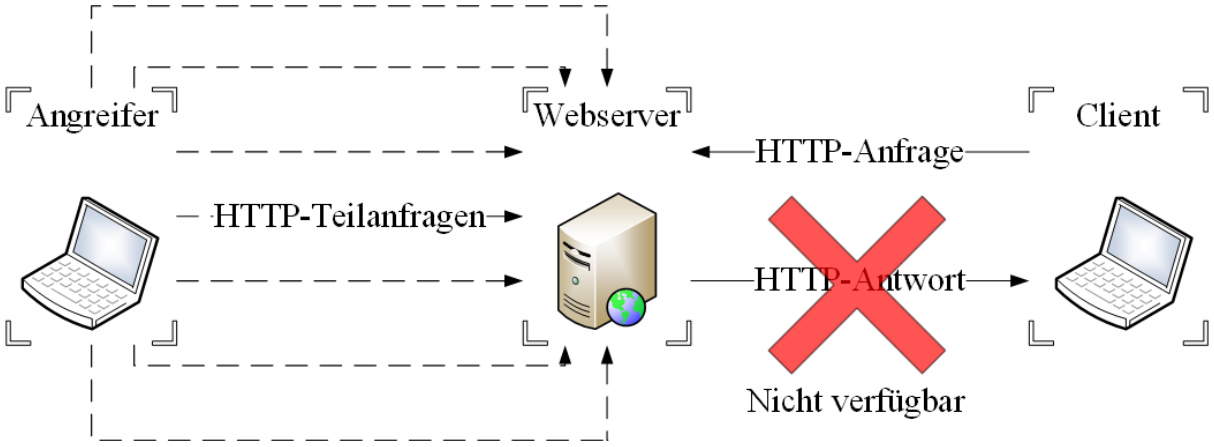

Abbildung 4-11 - DoS durch HTTP-Teilanfragen an Webserver

4.5.2 Angriffsmöglichkeit B: Der DNS-Server

In der Abbildung 4-1 wird die Schwachstelle E ausgenutzt. Die Verfügbarkeit des DNS-Servers soll eingeschränkt werden.

Der DNS-Server übersetzt die Webadresse einer Website in die IP-Adresse. Wird der DNS-Server, wie bereits in Abschnitt 2.2.5 beschrieben, mit fehlerhaften UDP Paketen überhäuft oder durch das genannte SYN-Flooding attackiert, ist für den Client die Website nicht mehr erreichbar, da er nicht mehr automatisch zur korrekten IP-Adresse weitergeleitet wird. Dieser Angriff ist nur relevant, wenn die Schüler über eine Domain oder auf die Website zugreifen.

4.5.3 Angriffsmöglichkeit C: Der Router

In der Abbildung 4-1 wird die Schwachstelle D verwendet. Sobald die Funktionalität des Routers gestört ist, wird die Kommunikation im WLAN eingeschränkt.

Der Raspberry Pi benutzt DHCP für die Vergabe von IP-Adressen an seine Clients. Allerdings sind die verfügbaren IP-Adressen begrenzt und mittels des Association-Flooding werden alle verfügbaren IP-Adressen durch Anfragen von gefälschten MAC-Adressen blockiert.

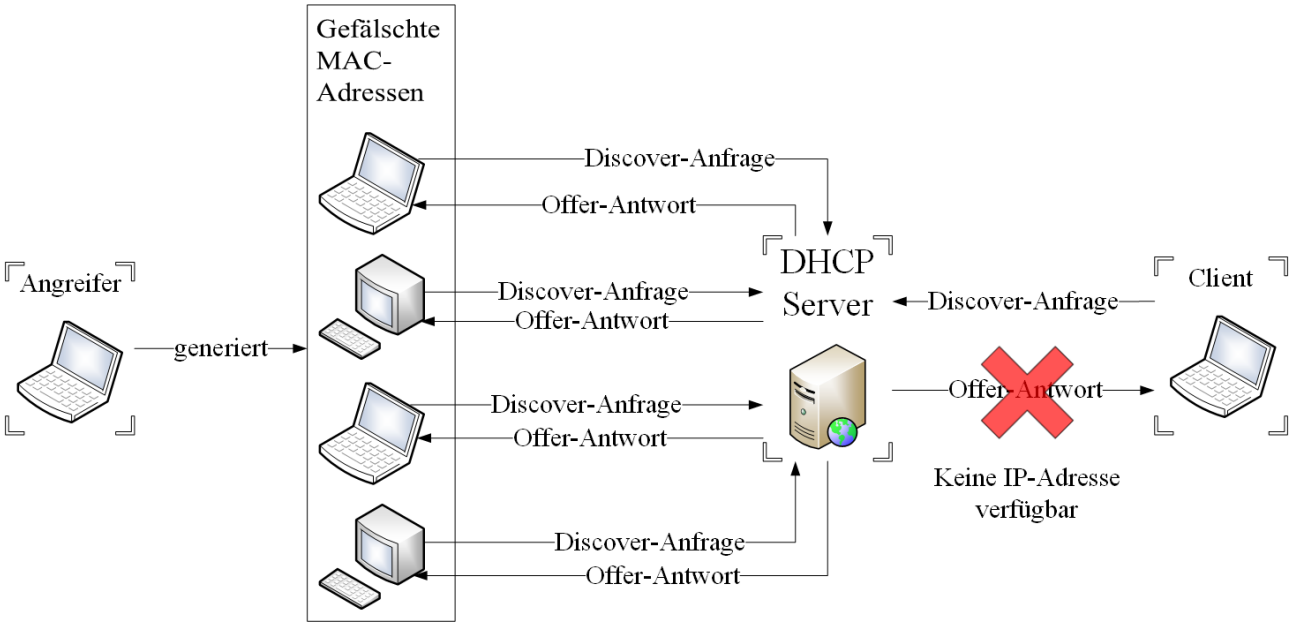

Abbildung 4-12 - DoS Attacke auf einen DHCP Server

Wie in Abbildung 4-12 dargestellt, sendet das Programm die im DHCP Protokoll definierten Discover-Nachrichten. Der Server wird daraufhin mit einer Offer-Nachricht antworten und dem Client eine IP-Adresse anbieten. Sobald der Angreifer genügend Discover-Nachrichten von unechten MAC-Adressen gesendet hat, um alle IP-Adressen zu belegen, ist DHCP überlastet. Reguläre Clients haben somit keinen Zugriff mehr auf das Netzwerk und der Datenverkehr ist gestoppt. Die Verfügbarkeit des Dienstes ist beeinträchtigt. In der Regel können 256 IP-Adressen in einem Netzwerk vergeben werden.

4.5.4 Realisierung des Angriffs auf einen Webserver

Von den vorgestellten Szenarien wird der Denial-of-Service-Angriff auf den Webserver genauer untersucht. In diesem Fall wird der Webserver auf dem Raspberry Pi 3 Model B gehostet. Um die Wirksamkeit des Angriffes zu demonstrieren, wird als Angreifer ein identischer Rechner verwendet. Demnach wird genau die gleiche Hardware eingesetzt und keiner der beiden Rechner hat einen Leistungsvorteil.

Der Angriff wird über das Programm `Slowloris`, welches die Anwendungsschicht ausnutzt, durchgeführt. Nach etwa 100 geöffneten Verbindungen vom Angreifer zum Webserver, ist die Website nicht mehr erreichbar. Die spannendere Tatsache ist, dass der Angreifer bis zu 3000 Verbindungen offenhalten kann. Demnach ist es nicht nur möglich mit identischer Hardware

den Webserver erfolgreich zu blockieren, sondern auch mit einer leistungsschwächeren Hardware. Wird auf die Ergebnisse ein simpler Dreisatz angewendet, so kann unter Umständen ein bis zu 30-mal leistungsstärkeren Webserver wirkungsvoll attackiert werden. Dieser Effekt wird durch die Ausnutzung der Anwendungsschicht erreicht. Mit dieser simplen Attacke kann ein einzelner Rechner bereits großen Schaden anrichten. Es muss nicht erst ein aufwändiges Botnetz erstellt und kontrolliert werden.

4.5.5 Lerninhalte und mögliche Abwehrmaßnahmen

Anhand des durchgeführten Szenarios kann den Schülern die grundlegende Funktion und Wirkung eines DoS-Angriffs erläutert werden. Außerdem können die Motive für solche Angriffe wie zum Beispiel Sabotage, Manipulation oder Erpressung verdeutlich werden.

Um den Webserver besser gegen solche Angriffe zu schützen, stehen verschiedene Ansätze zur Verfügung. Allerdings löst keiner dieser Ansätze das Problem endgültig. Zuerst könnte die maximale Anzahl an offenen Verbindung, die gehalten werden können, erhöht werden. Sinnvoller erscheint es allerdings, die maximale Anzahl an Verbindungen von einer IP-Adresse zu beschränken, denn bei `Slowloris` wird der Angriff von einem Rechner mit der gleichen IP-Adresse ausgeführt. Generell ist dieser Verkehr von einer IP-Adresse nicht beschränkt, denn in heutigen Netzwerken versteckten sich in der Regel mehrere Geräte hinter einer IP-Adresse wie beispielsweise im Heimnetzwerk. Dort sind viele Geräte beim WLAN-Router angemeldet, die alle gleichzeitig auf die gleiche Website zugreifen können, ohne dass es sich um einen DoS-Angriff handelt. Der letzte Ansatz, um den Webserver besser vor diesem Angriff zu schützen, verringert die Verbindungsdauer jedes Clients. Die unvollständigen HTTP-Anfragen werden durch neue Teilanfragen aktualisiert, damit sie nicht durch den Server aufgrund eines Timeouts geschlossen werden. Wenn diese Timeoutzeit verkleinert wird, werden die Verbindungen zügiger geschlossen und die Effektivität des Angriffs verringert. Der Angreifer könnte mit kürzeren Intervallen, in denen er neue Teilanfragen sendet, reagieren. Allerdings würde das ebenfalls mehr Rechenleistung beanspruchen und damit die Effektivität verringern.

4.6 Exploit

Dieser Angriff soll mittels eines Exploits die Kontrolle über das gesamte Betriebssystem des Raspberry Pi erlangen. Um eine rückwärtsgerichtet TCP-Verbindung zu etablieren, müssen sich der Angreifer und das Zielsystem beide im lokalen Netzwerk befinden oder über eine Internetverbindung verfügen.

In der Abbildung 4-1 werden die Schwachstelle A und B ausgenutzt. Der Benutzer öffnet zwar freiwillig eine infizierte Software und löst somit den Angriff aus, allerdings wäre dieser nicht ohne Schwachstellen in den Programmen oder dem Betriebssystem des Computers möglich.

4.6.1 Vorbereitung des Exploits

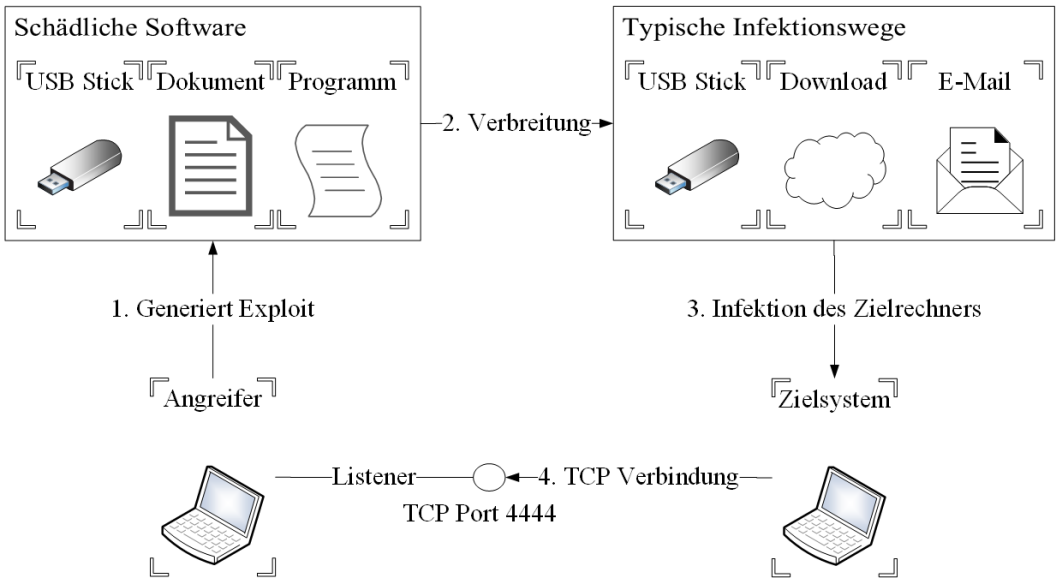

Abbildung 4-13 - Darstellung eines Exploits

Wie in der Abbildung 4-13 dargestellt, erstellt der Angreifer ein schädliches Programm, ein Dokument oder alternativ einen USB Stick. Das USB Protokoll definiert bereits Geräteklassen, daher können USB Sticks als Tastaturen deklariert werden und die Schadsoftware vollautomatisiert nach dem Einstecken in einen Rechner ausgeführt werden. Als Payload wird hier eine rückwärtsgerichtete TCP Verbindung benutzt. Die Schadsoftware öffnet demnach eine TCP Verbindung auf einem selten genutzten Port zu der IP-Adresse des Angreifers. Das Problem bei diesem Angriff ist, dass der Angreifer das Opfer dazu bringen muss, die Datei auf dem Zielsystem auszuführen bzw. zu öffnen. Zur Demonstration wird eine Schadsoftware, die unabhängig vom Betriebssystem über einen Implementierungsfehler in einer Anwendung eine rückwärtsgerichtete TCP-Verbindung öffnen kann, erstellt. Über diese soll dann eine Shellverbindung etabliert werden und wichtige Daten aus dem Raspberry Pi ausgelesen oder das System gesteuert werden. Für die Opfer ist das Problem eines solchen Angriffs, dass es ihnen womöglich nicht einmal auffällt, dass sie die Kontrolle verloren haben. Der Angreifer kann auf diesem Wege eine dauerhafte Verbindung etablieren, die sich immer wieder zu ihm aufbaut, sobald das Opfer sich mit dem Internet verbindet. Sowohl im privaten als auch im geschäftlichen Umfeld können hier Daten ausgespäht, Daten manipuliert, weitere Netzwerkteilnehmer infiziert oder ganze Rechner zerstört werden.

4.6.2 Durchführung des Exploits

Im Folgenden soll der im Abschnitt 4.6.1 erläuterte Exploit durchgeführt und seine korrekte Funktion validiert werden. Zur Erstellung der infizierten Datei wird das `Metasploit Framework` verwendet. Das Framework befasst sich hauptsächlich mit der Erstellung und Ausführung von Exploits gegen Zielrechner.

Um einen ausführbaren Exploit zu erstellen, ist es wichtig das Zielsystem zu kennen, denn das Opfer muss die Datei öffnen oder das Programm ausführen können. Das Zielsystem ist ein Raspberry Pi mit dem Betriebssystem `Raspbian Jessie`. Am besten eignen sich plattformunabhängige Programme. Aus diesem Grund soll ein Python Programm erstellt werden, welches den Exploit durchführt. Der Payload beinhaltet demnach ein Python Programm, das eine rückwärtsgerichtete TCP Verbindung zum Angreifer öffnet.

```
Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (python/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.1.18     yes       The listen address
   LPORT  4444             yes       The listen port
```

Abbildung 4-14 - Einstellungsoptionen für einen Exploit

In der Abbildung 4-14 sind die eingestellten Optionen für den gewählten Exploit abgebildet. In der ersten Zeile sind die Moduloptionen erläutert. Diese bedeuten, dass ein Exploit, der unabhängig vom Betriebssystem ist, erstellt werden soll. Ein Handler bezeichnet im Bereich der Informatik eine Rückruffunktion. Sobald das Programm ausgeführt wird, meldet sich der Handler beim Angreifer. Die Optionen für den Payload spezifizieren die oben genannten Einstellungen. Das Zielprogramm ist Python und über das Programm wird eine rückwärtsgerichtete TCP Verbindung geöffnet. Das Programm ermöglicht über eine SSL-Verbindung die Steuerung des angegriffenen Computers. Abschließend sind noch die Einstellungen LHOST und LPORT zusehen. LHOST bezeichnet die IP-Adresse, zu der die rückwärtsgerichtet TCP Verbindung aufgebaut werden soll. Die hier eingetragene IP-Adresse 192.168.1.18 bezeichnet demnach die Adresse des Angreifers im lokalen Netzwerk des Opfers.

Über LPORT wird lediglich der gewünschte TCP Port eingestellt. Es ist wichtig, einen nicht oder kaum benutzen Port zu benutzen. In diesem Beispiel wird der Port 4444 verwendet.

Nachdem nun die Einstellungen für den Angriff getroffen wurden, kann dieser erstellt werden. Das Opfer zum Ausführen des Programms zu verleiten, ist die schwierigste Aufgabe dieses Angriffs. In diesem Demonstrationsbeispiel wurde das Programm über einen USB Stick auf das Zielsystem übertragen und daraufhin ausgeführt. Der Angreifer muss natürlich auf eine Reaktion des Handlers warten. Reagiert der Angreifer nicht auf den Handler, kann er die Kontrolle über das Betriebssystem nicht übernehmen.

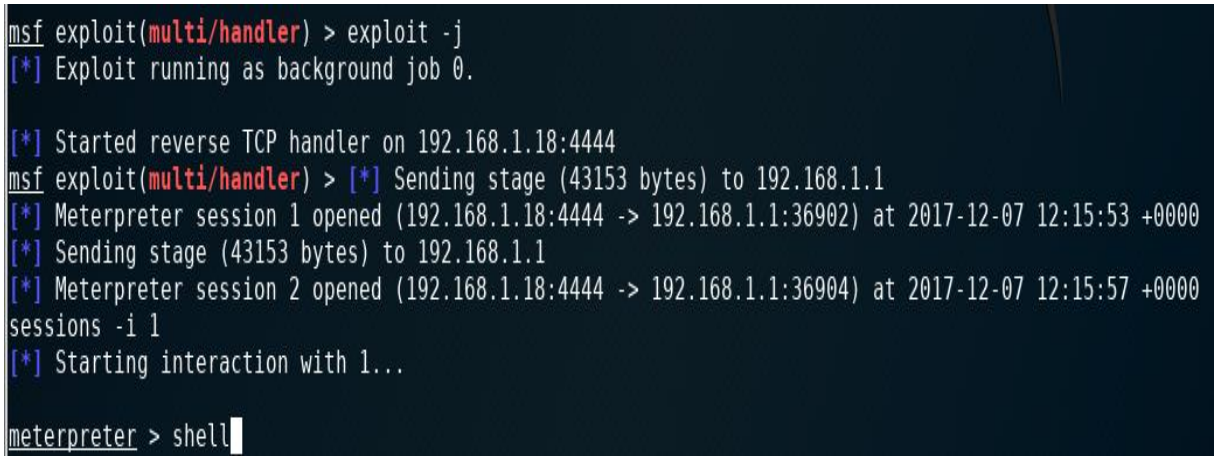

Abbildung 4-15 - Öffnen der TCP Verbindung nach erfolgtem Exploit

Die Abbildung 4-15 zeigt die Antwort des Handlers an den Angreifer. Dieser wartet im Hintergrund auf eine Reaktion vom Handler auf dem vorher spezifizierten Port 4444. Sobald das Opfer die schädliche Software ausführt, erhält der Angreifer eine Nachricht von der IP-Adresse 192.168.1.1. Diese IP-Adresse bezeichnet in diesem Angriff das Opfer. Das Programm bietet daraufhin direkt zwei Sitzungen an, die geöffneten werden können. In diesem Beispiel wird mit dem Befehl „`sessions -i 1`" die erste angebotene Sitzung geöffnet. In der letzten Zeile wird mit dem Befehl „`shell`" über das Programm eine Shell-Verbindung zum Opfer etabliert.

Nachdem diese Verbindung etabliert wurde, hat der Angreifer die Kontrolle über das Betriebssystem. Der Angreifer kann jetzt alle Systemdaten auszulesen, Verzeichnisse durchsuchen, ein Spionageprogramm platzieren oder auch Steuerbefehle absetzen. Das angegriffene Gerät kann also auch zum Neustart oder Herunterfahren gezwungen werden. Da der Angriff einen nicht verwendeten TCP Port benutzt, bekommt das Opfer in der Regel nicht mit, dass es auf einen Social-Engineering-Angriff hereingefallen ist. Der Bildschirm verändert sich nicht und somit kann der Angriff nur durch eine gezielte Analyse des Netzwerkverkehrs noch aufgedeckt werden.

4.6.3 Lerninhalte

Effektiven Schutz gegen diesen Angriff aus dem Bereich des Social Engineering bietet der gesunde Menschenverstand. Die Quelle einer Datei sollte stets auf ihre Vertrauenswürdigkeit geprüft werden. Mails von Fremden mit angehangenen Dokumenten, Downloads von unbekannten Websites oder gefundene USB-Sticks stellen immer eine potentielle Gefahrenquelle dar. Aus diesem Grund sollte das Ausführen oder Öffnen der Dateien und Dokumente vermieden werden. Eine weitere Abhilfe stellen aktualisierte Virenprogramme dar. Diese kennen in den aktuellsten Versionen derzeit bekannte Sicherheitslücken und erkennen diese. Das Virenprogramm scannt meistens alle fremden Dateien und stuft diese nach ihrem Sicherheitsrisiko ein. Sollte eine infizierte Datei erkannt werden, dann werden die betroffenen Dateien in der Regel direkt vom Virenprogramm gelöscht. Im Schulbereich kann der in Abschnitt 4.6.2 durchgeführte Exploit erläutert werden oder beispielsweise ein USB Stick mit infizierten Dokumenten, die ein Pop-Up-Fenster öffnen, ausgeteilt werden.

4.7 Zusammenfassung

In der Tabelle 4-1 wurden abschließend alle durchgeführten Angriffe zusammengefasst. Die für jeden Angriff notwendige Soft- sowie Hardware ist aufgelistet und der jeweilige Angriff kurz beschrieben.

Bezeichnung des Angriffs	**Software**	**Hardware**	**Kurzbeschreibung**
WPA2-Wörterbuch-Attacke	• Aircrack	• Raspberry Pi	Das Passwort eines WPA2 verschlüsselten Netzwerks wird durch Ausprobieren herausgefunden.
WPA2-Phishing-Attacke	• Fluxion	• Raspberry Pi • WLAN-Dongle	Das Passwort eines WPA2 verschlüsselten Netzwerks wird durch ein identisches Netzwerk des Angreifers, an welches der Nutzer das Passwort selbst verrät, herausgefunden.
Man-in-the-Middle	• Ettercap • Wireshark	• Raspberry Pi	Die Kommunikation zwischen Netzwerkteilnehmern wird über den Angreifer umgeleitet. Dieser kann die Daten mitlesen und manipulieren.
Denial of Service	• Slowloris	• Raspberry Pi	Die Verfügbarkeit eines Dienstes wird blockiert. In diesem Fall wird ein Webserver lahmgelegt.
Exploit	• Metasploit Framework	• Raspberry Pi	Über einen Implementierungsfehler wird eine rückwärtsgerichtete TCP-Verbindung zum Angreifer geöffnet. Dadurch erlangt dieser die volle Kontrolle über das Betriebssystem.

Tabelle 4-1 - Zusammenfassung der durchgeführten Angriffe

4.8 Didaktische Anknüpfungspunkte

In diesem Abschnitt werden mehrere didaktische Anknüpfungspunkte für eine weitere pädagogisch-didaktische Ausarbeitung der Projekte aus dem Bereich Datensicherheit gegeben.

- Wie kann ein sicheres Passwort zum Schutz der eigenen Privatsphäre erstellt werden?
- Wie funktionieren Wörterbuch-Attacken und wieso schützen kurze Passwörter unzureichend?
- Gibt es einen Schutz vor Phishing?
- Wie leicht fallen die Schüler auf Social-Engineering-Attacken herein?
- Wie sicher sind öffentliche Netzwerke?
- Welche Verschlüsselungsstandards sollten beachtet werden?
- Was ist der Unterschied zwischen HTTP und HTTPS?
- Was kann passieren, wenn eine infizierte Datei aus dem Internet heruntergeladen wird?
- Wie funktionieren Denial-of-Service-Angriffe?
- Was sind die Motive für Denial-of-Service-Angriffe?

Kapitel 5

Schülerprojekte im Bereich Künstliche Intelligenz

Die im Abschnitt 3.2 entwickelten Ansätze zur Realisierung einer künstlichen Intelligenz sollen hinsichtlich des Einsatzes im Schulunterricht analysiert werden. Durch die Realisierung und Validierung eines ausgewählten Ansatzes soll der Nutzen für das Projekt letsgoING bestätigt werden.

5.1 Auswahl einer KI-Anwendung

Projekt	Hardware	Kurzbeschreibung
ArduRover	• ArduRover • SD-Karten-Modul • Raspberry Pi	Fahrzeug folgt durch eine Zweipunktreglung einer schwarzen Linie. Die Zweipunktregelung wird durch ein neuronales Netz ersetzt.
Klassifizierung von Körpern	• Förderband • Infrarotsensor • Raspberry Pi	Ein Infrarotsensor rastert Körper, die auf einem Förderband stehen, ab. Das neuronale Netz unterscheidet durch diese Daten die Körper.
Klassifizierung von Zuständen	• Verteilte Messstationen • Raspberry Pi	In den Klassenzimmern werden über verteilte WLAN-Module verschiedene Zustände klassifiziert.

Tabelle 5-1 - Übersicht über die möglichen Projekte im Bereich künstliche Intelligenz

Eine KI-Anwendung kann nur im Rahmen des Projektes letsgoING eingesetzt werden, wenn die Software kostenlos, frei verfügbar und plattformunabhängig ist. Außerdem muss die Ausführung der künstlichen Intelligenz auf dem Arduino Uno oder einem zentralen Raspberry Pi 3 Model B erfolgen. Des Weiteren darf die Komplexität nicht zu hoch sein, denn das neuronale Netz muss auch auf dem Raspberry Pi angelernt werden können. Es ist wichtig, dass Anknüpfungspunkte an die bestehenden Lerninhalte vorhanden sind. Die Tabelle 5-1 fasst die möglichen Projekte zusammen. Diese sollen im Folgenden hinsichtlich ihres Einsatzes im

Schulunterricht analysiert werden. Alle drei neuronalen Netze können mit dem Programm `TensorFlow` in der Programmiersprache Python erstellt werden.

Die verschiedenen Anwendungsszenarien sind alle im Rahmen des Projektes letsgoING anwendbar. Allerdings ist die Klassifizierung der Körper für den Unterricht eher ungeeignet. Die notwendige Hardware müsste vollständig neu gekauft und für die Schulen aufbereitet werden. Außerdem ist das neuronale Netz vollständig programmiert und die notwendigen Trainings- und Testdaten sind bereits verfügbar. Aus diesem Grund wäre der Aufwand für die Software sehr gering, jedoch soll gerade die Erstellung eines neuronalen Netzes in dieser Arbeit untersucht werden. Durchaus könnte das vorhandene Netz optimiert werden, allerdings erzielt es einen sehr hohen Leistungsindex mit bis zu 95% richtigen Entscheidungen. Zu optimieren wären die verwendeten Rechner, denn für den Einsatz in der Schule wäre es wünschenswert, nicht zwei verschiedene Systeme verwenden zu müssen. Dies steigert den Aufwand und die Komplexität deutlich.

Die Datenanalyse durch räumliche verteilte WLAN-Module mit einem zentralen Broker bietet einen spannenden Anwendungsfall für das neu entwickelte Modul aus dem Bereich Internet der Dinge. Es bietet eine interessante Kombination aus dem Aufbau einer IP-Kommunikation mit einer anschließenden Analyse der Daten. Es muss allerdings noch getestet und überprüft werden, ob es mit den im Projekt letsgoING verfügbaren Sensoren möglich ist, die gewünschten Vorhersagen zu treffen. Wenn das Konzept mit diesen Sensoren funktioniert, ist keine weitere Hardware notwendig. Der große Vorteil an diesem Projekt ist, dass die Berechnung und das Anlernen des Netzes vollständig auf einem zentralen, leistungsstarken Rechner ausgeführt werden kann. Das Problem in diesem Fall ist definitiv die Aufnahme von Trainings- und Testdaten. Entweder es wird über Bildverarbeitung ausgewertet, welcher Zustand gerade im Raum herrscht und wie viele Personen anwesend sind oder die jeweilen Zustände müssten manuell sehr genau protokolliert werden. Da Bildverarbeitung aufgrund von Datenschutzbestimmungen nicht realisierbar ist, müssten die Zustände ständig protokolliert werden. Außerdem wird das Ergebnis von vielen weiteren Faktoren wie beispielsweise der Größe des Raums und der Belüftung abhängig sein. Ein denkbarer Ansatz wäre das Lernen ohne Lehre mit Clustering-Methoden. Allerdings sind für diese Verfahren sehr leistungsstarke Rechner notwendig. Die Aufnahme der Trainings- und Testdaten oder die Analyse, ob ein Clustering möglich ist, müssten im Rahmen eines längerfristigen Projekts in Zusammenarbeit mit den Schulen geprüft werden.

Das Projekt ArduRover wird bereits seit einiger Zeit erfolgreich an den Schulen eingesetzt und die verantwortlichen Lehrer haben schon sehr viel Erfahrung mit dem Modul. Des Weiteren ist

die Hardware vollständig vorhanden und für den Unterricht aufbereitet. Aus diesem Grund sind die Kosten sehr gering. Außerdem kann hier das neuronale Netzwerk vollständig programmiert werden und die effizienteste Konfiguration implementiert werden. Darüber hinaus ist es denkbar, die Motoren über Handregler oder über eine App anzusteuern und damit die Trainings- und Testdaten zu generieren. Hier würde der Effekt, dass sorgfältig aufgenommene Daten zu besseren Ergebnissen führen, verstärkt werden. Alternativ werden die Daten mit dem von den Schülern erstellten Programmen aufgenommen. Diese Daten können vom Arduino Uno direkt auf eine SD Karte in dem benötigten Format abspeichert werden. Allerdings muss das Training des neuronalen Netzes ebenfalls auf einem zweiten leistungsstärkeren Rechner stattfinden. Abschließend muss getestet werden, ob das neuronale Netz auf dem Arduino ausgeführt werden kann.

Projekt	Hardware	Vorteile	Nachteile
ArduRover	• ArduRover • SD-Karten-Modul • Raspberry Pi	• Hardware vorhanden und in Schulen bekannt • Erstellung der Datensätze schnell möglich	• Zwei Rechner
Klassifizierung von Körpern	• Förderband • Infrarotsensor • Raspberry Pi	• Programm vorhanden • Gute Dokumentation • Datensätze vorhanden	• Förderband und Infrarotsensor nicht vorhanden • Zwei Rechner
Klassifizierung von Zuständen	• Verteilte Messstationen • Raspberry Pi	• Zentraler Rechner • Clusteringverfahren evtl. möglich	• Aufnahme der Datensätze aufwendig

Tabelle 5-2 - Vergleich der möglichen Projekte im Bereich künstliche Intelligenz

In der Tabelle 5-2 werden die drei Projekte mit ihren Vor- und Nachteilen sowie der notwendigen Hardware verglichen.

Vor dem Hintergrund, dass der ArduRover bereits erfolgreich eingesetzt wird, die Hardware vollständig vorhanden ist und die Trainings- sowie Testdaten am schnellsten erstellt werden können, soll dieses Projekt im Rahmen dieser Bachelorarbeit realisiert werden.

5.2 Erstellen der Trainings- und Testdaten

Zum Anlernen des neuronalen Netzwerks müssen sowohl Trainings- als auch Testdaten erstellt werden, da die künstliche Intelligenz mit einem Lernverfahren mit Lehrer trainiert wird. Das Lernverfahren wurde im Abschnitt 2.3.3 erläutert. Das wichtigste ist, dass die beiden Datensätze weder die gleichen sind, noch sich zu stark ähneln. In diesem Anwendungsfall wird das gleiche Streckenprofil mit der klassischen Zweipunktreglung abgefahren und währenddessen jeweils zwei Dateien erstellt. Zur Unterscheidung der Datensätze wird die Fahrtrichtung und die Beleuchtung der Strecke geändert. Die erste Datei enthält die Werte der Lichttaster und die zweite Datei die Werte der Schrittmotoren. Um eine weitere Verarbeitung zu vereinfachen, wird jedes Wertepaar nur durch ein Leerzeichen getrennt in eine Zeile geschrieben. Nach diesem Schritt sollten vier Dateien vorhanden sein. Diese werden direkt vom Arduino Uno über SPI auf eine SD-Karte geschrieben. Ein notwendiges SD-Karten-Modul für den Arduino Uno muss bereitgestellt werden. Die erstellten Datensätze können bei einer fehlerlosen Fahrt ungefiltert verwendet werden.

Eine Alternative zu dem im Schulprojekt verwendeten Programm stellt die direkte Ansteuerung der Motoren dar. Die Gleichstrommotoren könnten direkt über zwei Potentiometer, über Handregler oder mit einer App über das Smartphone gesteuert werden. Der Arduino nimmt hierbei die Messdaten automatisiert auf. Der Lerneffekt ist deutlich größer, da zur Aufnahme nicht schon ein Programm notwendig ist. Problematisch bei diesem Verfahren ist allerdings die Filterung der Messdaten, denn die Strecken müssen ohne Fahrfehler abgefahren werden. Außerdem ist es schwierig die Aufnahme zu starten und zu beenden. Unter Umständen müssen die ersten und letzten Messdaten verworfen werden.

Wie bereits erwähnt ist die Aufnahme der Messdaten mit dem Programm deutlich einfacher. Sobald der Arduino gestartet wird, fährt dieser auf der Linie und erfasst die Messdaten. Für die Messdatenspeicherung mit manuellem Fahrer muss auf dem Arduino eine zusätzliche Regelung erfolgen. Diese soll starten, sobald die Potentiometer die Ausgangsstellung verlassen hat und dementsprechend auch wieder stoppen.

Das Problem an der ersten Variante ist, dass in dem Programm nur zwei Zustände berücksichtigt werden. Wenn die Differenz der beiden Lichtsensoren einen Wert unter- bzw. überschreitet wird entsprechend eine Links- oder eine Rechtskurve mit einer festen Geschwindigkeit gefahren. Hier ist die zweite Variante deutlich besser, denn die Qualität des neuronalen Netzes hängt direkt mit den Fahrkünsten des Fahrers zusammen. Dieser kann unabhängig von den Sensorwerten und mit verschiedenen Geschwindigkeiten die Linie

abfahren. Allerdings ist es fraglich, ob die Datensätze zum Anlernen genutzt werden können, denn der Fahrer fährt ohne Berücksichtigung der Sensorwerte. Aus diesem Grund ist es sinnvoll, die Sensorwerte der Lichttaster zu visualisieren. Durch diese Visualisierung kann der Fahrer den ArduRover in Abhängigkeit der Sensorwerte steuern.

Die Logik hinter den Sensorwerten erkennt dann das neuronale Netz und kann voraussichtlich die Motorgeschwindigkeit dynamisch anpassen. Dieser Vorgang der Generalisierung wurde bereits im Abschnitt 2.3.2 erläutert. In dieser Arbeit soll die automatisierte Aufnahme der Daten mittels des Programmes getestet werden.

Ein geeignetes Verhältnis zwischen Trainings- und Testdaten ermöglicht einen effizienten Lernzyklus für das neuronale Netz. Aus diesem Grund werden, wie häufig in der Literatur empfohlen, 700 Trainingsdaten sowie 300 Testdaten erstellt. Ein gutes Verhältnis wird meistens auf etwa 70:30 beziffert. [21]

5.3 Auswahl der Struktur des neuronalen Netzwerkes

Die in Abschnitt 5.1 ausgewählte Anwendung soll auf einem Arduino Uno ausgeführt werden können. Das Training des Netzwerkes soll auf einem Raspberry Pi 3 Model B stattfinden, daher eignen sich sehr kleine neuronale Netze mit maximal einer versteckten Schicht.

Die einfachste Form eines neuronalen Netzwerkes ist das sogenannte Perzeptron. Bei diesem sind die Eingabevektoren direkt mit den Ausgabevektoren verknüpft. Dieses eignet sich für sehr simple Aufgaben wie zum Beispiel ein 2-Bit-Addierer. Des Weiteren können die logischen Funktionen AND, OR und NOT mit diesem Netzwerk realisiert werden. Ein Perzeptron bietet den geringsten Rechenaufwand sowohl in der Ausführung als auch beim Anlernen. Allerdings wird dieses nicht die notwendige Funktionalität für die ausgewählte Anwendung erfüllen können. Es wird auch als einschichtiges Feedforward-Netz bezeichnet. Feedforward bedeutet, dass die Schichten nur mit der nächsthöheren Schichte verbunden sind und keine Rückkopplungen zu tieferen Schichten enthalten. [22]

Die für den Anwendungsfall geeignetere Form ist das mehrschichtige Feedforward-Netz. Das Netzwerk hat zwischen der Eingabe- und der Ausgabeschicht beliebig viele versteckte Schichten. Bei der ausgewählten Anwendung genügt eine versteckte Schicht. Diese kann beliebig viele Neuronen aufweisen und bietet somit deutlich mehr Funktionalität. Außerdem ist das Netzwerk mit wenigen Neuronen in der versteckten Schicht immer noch sehr kompakt und auf einem Raspberry Pi 3 Model B anlernbar.

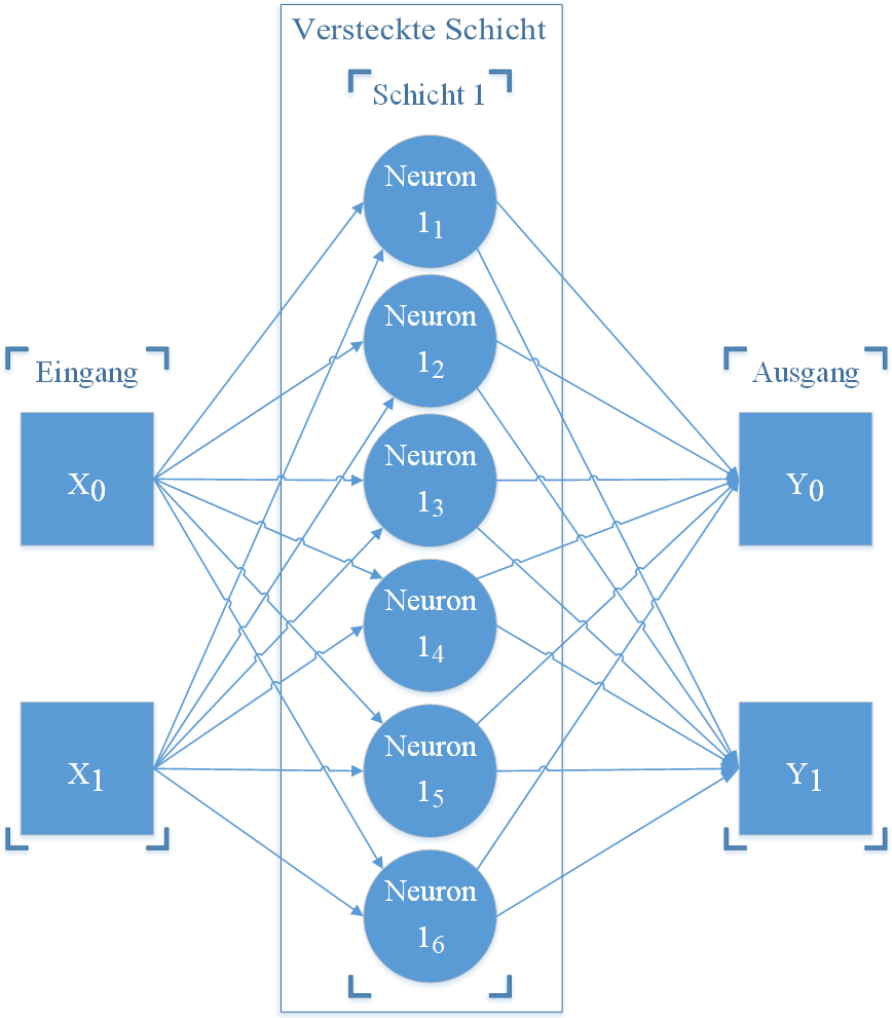

Abbildung 5-1 - Ausgewähltes Feedforward-Netzwerk

In der Abbildung 5-1 ist das ausgewählte Feedforward-Netzwerk dargestellt. Die Eingangsparameter X_0 X_1 sind wie bereits spezifiziert die Werte der Lichttaster und die Ausgangsparameter Y_0 Y_1 die Ansteuerungswerte für die Gleichstrommotoren. Die versteckte Schicht wurde mit sechs Neuronen besetzt, da es für die Anwendung der Klassifizierung von verschiedenen Körpern ebenfalls ausreichend war und dort ein Leistungsmaß von 95% erzielt wurde. Im weiteren Verlauf der Arbeit wird das Feedforward-Netzwerk noch mit weniger und mehr Neuronen in der versteckten Schicht hinsichtlich des Leistungsmaßes sowie der Rechenzeit überprüft.

5.4 Training des neuronalen Netzwerkes

Nachdem im Abschnitt 5.2 das Vorgehen zur Erstellung der Trainings- und Testdaten und im Abschnitt 5.3 die Struktur des Netzwerkes festgelegt wurde, müssen abschließend noch verschiedene Parameter für das Training des Netzwerks analysiert werden.

Zuerst muss die Initialisierung der Gewichte und Offsets festgelegt werden. Da keine vergleichbaren Projekte oder Erfahrungen zur Verfügung stehen, werden diese mit Zufallszahlen initialisiert. Einerseits bedeutet dies den geringsten Aufwand zu Beginn des Trainings, andererseits ist das Trainingsergebnis stark abhängig von den initialisierten Zufallszahlen. Dementsprechend wird jedes Mal, wenn das Netz trainiert wird, ein anderes Ergebnis herauskommen. Dabei werden sowohl sehr schlechte Leistungsmaß, als auch sehr gute Leistungsmaße erreicht. Nach mehreren Berechnungen ist das maximale Leistungsmaß für ein Netzwerk bekannt. Die berechneten Gewichte und Offsets sollten daraufhin von diesem sehr guten Leistungsmaß für die spätere Anwendung abgespeichert werden.

Als Aktivierungsfunktion für die Neuronen wird die Funktion Rectified Linear Unit ReLU eingesetzt.

Anschließend muss noch die Abweichung der berechneten Werte und der korrekten Werte bestimmt werden. Hierfür stehen in der Bibliothek `TensorFlow` verschiedene Möglichkeiten zur Verfügung. Da mittels einer Regression die Steuerungsparameter für die Gleichstrommotoren bestimmt werden sollen, wird die Abweichung über die mittlere quadratische Abweichung berechnet. Diese Funktion wird in `TensorFlow` als Kostenfunktion bezeichnet. Zur Nachbildung des Zweipunktreglers würde eine Klassifizierung genügen, allerdings kann mittels einer Regression die Geschwindigkeit der Motoren dynamischer an die aktueller Fahrsituation angepasst werden. Aus diesem Grund wird eine Regression zur Berechnung der Ausgangsparameter verwendet.

Wenn die Abweichung bestimmt ist, muss außerdem noch eine Optimierungsfunktion festgelegt werden, um die Abweichung nach jedem Durchlauf zu minimieren. Auch hier stellt `TensorFlow` verschiedene Funktionen zur Verfügung. Die am häufigsten verwendete Funktion ist der AdamOptimizer, der auch im bereits erklärten Projekt der c’t verwendet wird. Aus diesem Grund wird ebenfalls der AdamOptimizer verwendet.

Abschließend läuft das Programm in beliebig vielen Epochen über die kompletten Trainingsdaten. Nach jeder Epoche werden die Parameter des Netzes aktualisiert. Mit einer größeren Anzahl an Epochen kann das Ergebnis somit meistens verbessert werden. Die Anzahl der benötigen Epochen hängt stark von der Initialisierung der Parameter ab, daher können auch mit wenigen Epochen sehr gute Leistungsmaße erzielt werden. Des Weiteren muss die benötigte Rechenzeit betrachtet werden, die mit einer größeren Anzahl an Epochen sowie einem umfangreicheren Trainingsdatensatz ebenfalls steigt. In der Abbildung 5-2 wurde der Lernvorgang in `TensorFlow` als Ablaufdiagramm dargestellt. In diesem Beispiel wird das neuronale Netz mit einem Trainingsdatensatz der Größe 700 über 100 Epochen trainiert. Der

Lernprozess wurde in die einzelnen Schritte Initialisierung, Berechnung der Trainingsdaten, Auswertung der Ergebnisse und Bestimmung des Leistungsindex unterteilt. Nachdem die Initialisierung abgeschlossen ist, werden alle Trainingsdaten durchgerechnet und die Kostenfunktion bestimmt. Diese bestimmt die Abweichung der berechneten und korrekten Ergebnisse. Nachdem alle Trainingsdaten berechnet wurden, werden die Matrizen angepasst. Die Epoche wird erhöht und die Trainingsdaten erneut durchgerechnet. Sobald alle vorgegebenen Epochen gerechnet wurden, wird das neuronale Netz auf die Testdaten angewendet und somit der Leistungsindex bestimmt.

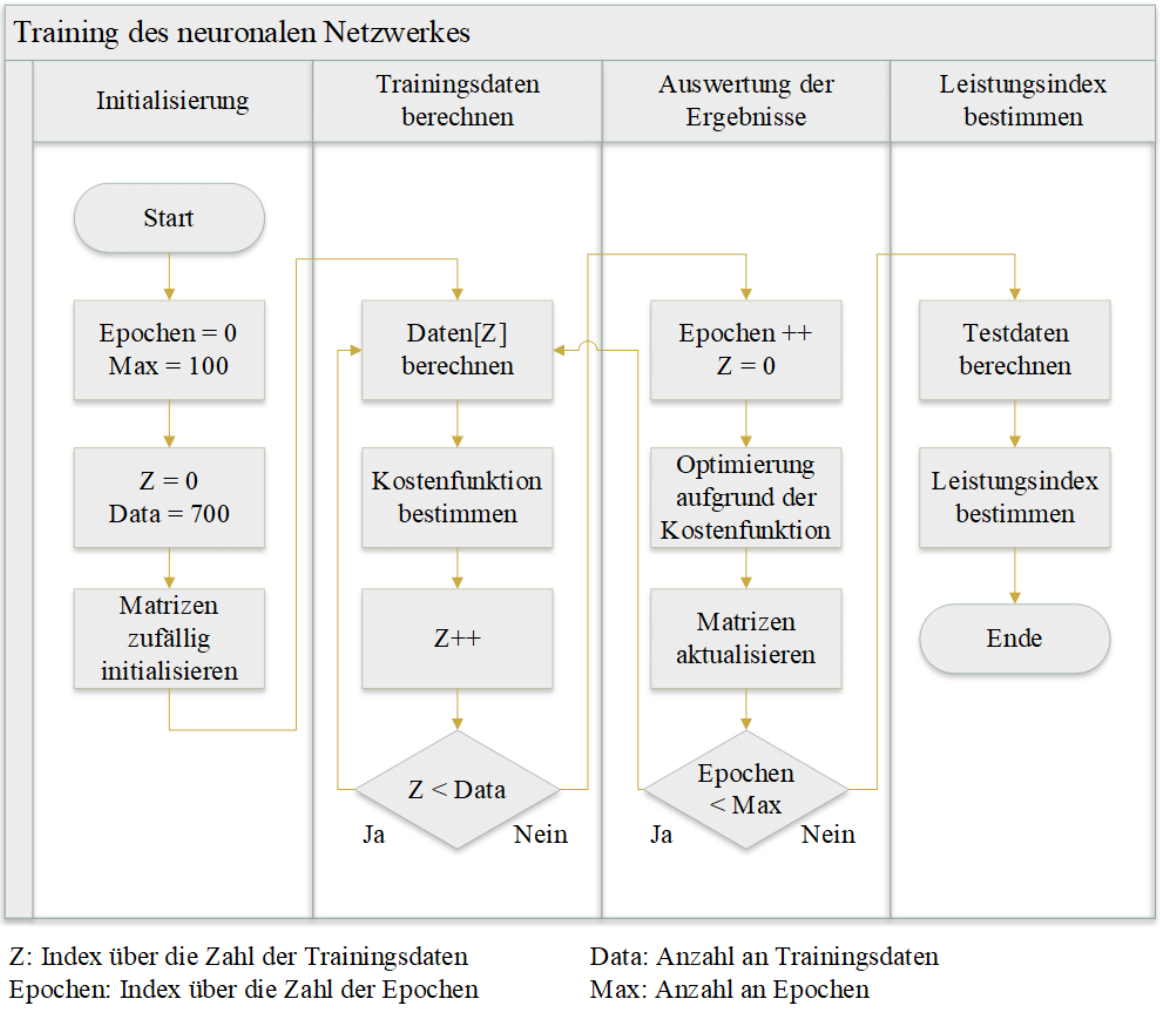

Abbildung 5-2 - Ablaufdiagramm Training eines neuronalen Netzes mit TensorFlow

5.5 Anpassung der Parameter des neuronalen Netzes

Die Parameter für das neuronale Netz wurden im Abschnitt 5.4 erläutert. Die Trainings- und Testdaten wurden mit einem Programm erstellt, dass die Lichttaster abfragt und mit einer Zweipunktregelung darauf reagiert. Als Kostenfunktion wird die mittlere quadratische Abweichung verwendet und als Optimierungsfunktion wird der AdamOptimizer mit einer

Standardlernrate von 0.001 eingesetzt. Als Aktivierungsfunktion wird die Funktion Rectified Linear Unit ReLU eingesetzt. Außerdem wurde das neuronale Netz mit sechs Neuronen in der versteckten Schicht klassifiziert. Die 700 Trainingsdaten und 300 Testdaten werden zu Beginn in 250 Epochen antrainiert. Diese Parameter sollen im Folgenden untersucht und unter Berücksichtigung der Anwendung verbessert werden.

Da für die Anwendung die Komplexität möglichst weit reduziert werden muss, wird zuerst die Anzahl der versteckten Neuronen untersucht. Für den Arduino Uno bedeutet jedes Neuron einen zusätzlichen Rechenaufwand. Aus diesem Grund muss die Komplexität möglichst weit reduziert werden. Außerdem ist die Generalisierung relativ simpel, daher darf die Komplexität des Netzes nicht so groß werden, damit das Netz die Trainingsdaten nicht einfach auswendig lernt und somit nicht mehr generalisiert. Dieses Problem wird im Bereich des maschinellen Lernens als Overfitting bezeichnet. Wenn Overfitting in einem neuronalen Netz auftritt, erzielt es zwar im Training gute Leistung, kann aber die Testdaten nur sehr schlecht berechnen.

Des Weiteren ist zu beachten, dass die Matrizen zufällig initialisiert werden und damit jeder Durchlauf ein anderes Ergebnis hervorbringt. Für die Anwendung ist allerdings nur der beste Fall von Bedeutung, da das Netz mehrmals angelernt werden und der optimale Durchgang gespeichert werden kann. Aus diesem Grund wird für jede Konfiguration nur das Optimum aus fünf Anlernvorgängen betrachtet.

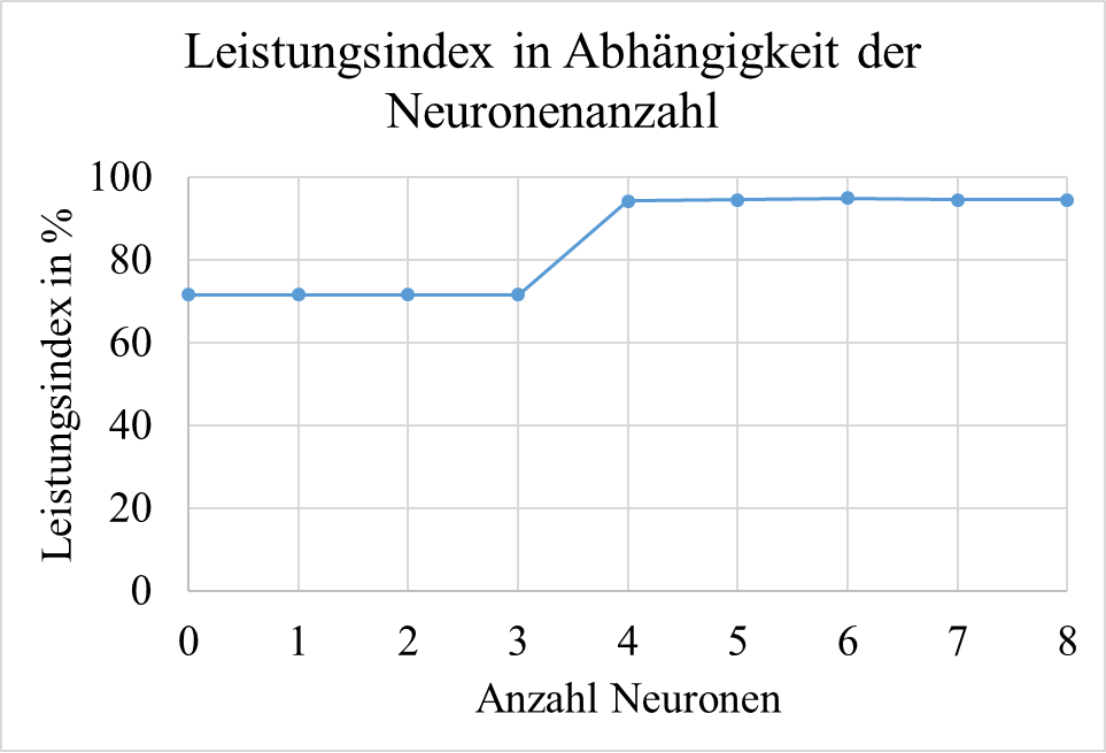

Abbildung 5-3 - Abhängigkeit zwischen Leistungsindex und der Anzahl an Neuronen

In der Abbildung 5-3 wird die Abhängig zwischen der Anzahl an Neuronen und des Leistungsmaßes deutlich. Die Genauigkeit wurde anhand der in Abschnitt 5.2 erstellten Trainings- und Testdaten berechnet. Von null bis drei Neuronen auf der versteckten Schicht liegt die Genauigkeit des neuronalen Netzes bei ungefähr 70%. Bei keinem Neuron wird diese Genauigkeit nur noch über einen Offset der Ausgabeschicht erreicht. Ab dem vierten Neuron erhöht sich die Berechnungsgenauigkeit deutlich auf über 90%. Allerdings bedeutet dies nicht, dass weitere Neuronen die Genauigkeit weiterhin signifikant erhöhen, denn mit fünf, sechs, sieben oder auch acht Neuronen lässt sich nur eine ähnliche Genauigkeit erreichen. Entgegen des im Abschnitt 5.3 ausgewählten Feedforward-Netzes mit sechs Neuronen in der versteckten Schicht, werden die Neuronen auf vier gekürzt, um für die spätere Applikation eine niedrigere Rechenlast zu gewährleisten.

Als nächstes soll die Anzahl der Epochen untersucht werden. Prinzipiell steht für das Training des Netzwerkes im Schulprojekt der Raspberry Pi 3 Model B zu Verfügung. Dieser könnte eine beliebige Anzahl an Epochen berechnen, allerdings bedeuten viele Epochen auch eine lange Trainingszeit. Da im Unterricht nicht eine Stunde auf ein brauchbares Ergebnis gewartet werden kann, muss eine sinnvolle Anzahl an Epochen untersucht werden. Dies geschieht nicht wie in der vorherigen Festlegung über das Optimum, sondern über den Durchschnitt von fünf Messungen. Der Durchschnitt ist sinnvoller, denn auch mit nur einer Epoche kann bei den zufälligen initialisierten Matrizen eine hohe Genauigkeit von über 90% erreicht werden, allerdings muss hierzu der Lernprozess häufig gestartet werden. Wenn hingegen der Durchschnitt betrachtet wird, ist die Wahrscheinlichkeit deutlich größer, dass bei den ersten Lernprozessen bereits sinnvolle Parameter bestimmt werden können und die Frustration in den Schulen geringer ausfällt. Allerdings kann mit einer kleinen Anzahl an Epochen den Schülern der Lernprozess besser erklärt werden, denn es kommt häufiger zu schlechten Ergebnissen. Ähnlich wie beim menschlichen Lernprozess lernen neuronale Netzwerke unterschiedlich schnell.

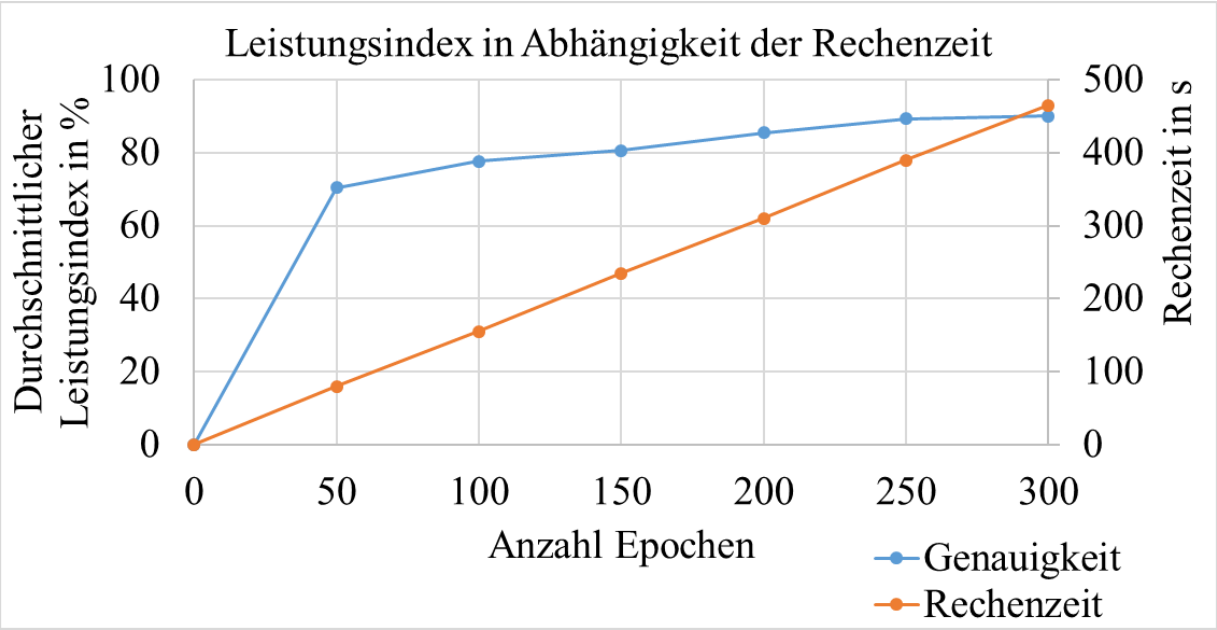

Abbildung 5-4 - Abhängigkeit zwischen Leistungsindex sowie Rechenzeit und Epochen

In der Abbildung 5-4 ist der Zusammenhang zwischen der Anzahl der Epochen und des Leistungsmaßes bei einem einschichtigen Feedforward-Netz dargestellt. Hierbei wird wie bereits beschrieben der durchschnittliche Leistungsindex betrachtet. Außerdem lässt sich erkennen, dass mit steigender Anzahl an Epochen ebenfalls der durchschnittliche Leistungsindex steigt. Allerdings ist die Rechendauer bei 300 Epochen auf dem Raspberry Pi 3 Model B mit knappen acht Minuten schon sehr lang. Die Anzahl von 200 Epochen bietet eine gute Balance zwischen durchschnittlicher Genauigkeit von knappen 90% und einer Rechenzeit von etwa fünf Minuten. Die Verbesserung von 200 zu 250 ist zwar noch deutlich zu erkennen, aber die Rechenzeit bei 250 Epochen ist mit etwa 390 Sekunden für den Einsatz im Unterricht bereits sehr lang. In diesem Fall ist die Rechenzeit das wichtigere Kriterium ist. Darüber hinaus ist auch mit sehr wenigen Epochen im Optimalfall ein sehr gutes Ergebnis möglich. Aus diesem Grund ist eine Anzahl von 200 Epochen folgerichtig. Bei der gewählten Anzahl an Epochen treten häufig Ergebnisse von über 90% auf, welche als sinnvolle Ergebnisse eingestuft werden können.

5.6 Anwendung auf dem Arduino Uno

Durch die Berechnung des neuronalen Netzes mithilfe von `TensorFlow` werden die Werte für die Matrizen der einzelnen Schichten berechnet. In dem gewählten Anwendungsfall mit einer versteckten Schicht entstehen so zwei Matrizen für die Gewichte und zwei Matrizen für die Offsets. Mit der ersten Matrizenmultiplikation und -addition werden die Eingangsparameter

auf die versteckten Neuronen abgebildet. Im zweiten Rechenschritt werden die versteckten Neuronen dann direkt auf die Ausgabeschicht abgebildet.

Um die vier Rechenoperationen auf dem Arduino Uno zu realisieren, wird die Bibliothek MatrixMath eingebunden. Mit dieser können die Funktionen Matrizenmultiplikation und Matrizenaddition auf dem Mikrocontroller ausgeführt werden. Die Messwerte der Lichttaster werden direkt in eine Matrix eingelesen. Auch die Ansteuerung der Motoren erfolgt direkt aus einer Matrix. Für eine optimale Rechenzeit wird das Programm auf seine elementarsten Funktionen reduziert. Die Sensoren werden ausgelesen, die Berechnungen durchgeführt und die Motoren angesteuert.

$$\begin{matrix} N_{11} & N_{12} & N_{13} & N_{14} \end{matrix} = \begin{matrix} X_0 & X_1 \end{matrix} \bullet \begin{matrix} W_{h1} & W_{h2} & W_{h3} & W_{h4} \\ W_{h5} & W_{h6} & W_{h7} & W_{h8} \end{matrix} + \begin{matrix} B_{h1} & B_{h2} & B_{h3} & B_{h4} \end{matrix}$$

$$\begin{matrix} Y_0 & Y_1 \end{matrix} = \begin{matrix} N_{11} & N_{12} & N_{13} & N_{14} \end{matrix} \bullet \begin{matrix} W_{o1} & W_{o2} \\ W_{o3} & W_{o4} \\ W_{o5} & W_{o6} \\ W_{o7} & W_{o8} \end{matrix} + \begin{matrix} B_{o1} & B_{o2} \end{matrix}$$

In den obenstehenden Formeln wird zuerst die Umrechnung der Eingangsparameter $X_0 \quad X_1$ auf die versteckte Schicht mit vier Neuronen $N_{11} \quad N_{12} \quad N_{13} \quad N_{14}$ dargestellt. Anschließend wird die versteckte Schicht auf die Ausgangsparameter $Y_0 \quad Y_1$ abgebildet. Die Gewichte W und Offsets B wurden als Variablen angegeben. Da das neuronale Netz bei jedem Anlernen andere Werte für diese Variablen berechnet, ist es nicht sinnvoll, diese anzugeben.

Außerdem werden zur Vereinfachung die von `TensorFlow` berechneten Gleitkommazahlen auf die zweite Nachkommastelle gerundet. Diese Rundung spart nicht nur Speicherplatz auf dem Arduino Uno, sondern beschleunigt auch die Berechnung der Ausgabewerte.

Das erstelle Programm berechnet die Ausgabewerte ausreichend schnell und kann somit auf die Gegebenheiten der Strecke reagieren. Der Arduino Uno bildet die Entscheidung korrekt auf das neuronale Netz ab und kann der Linie folgen. Durch die verwendete Regression werden die Ausgabewerte für die Motoren sogar differenzierter als bei der Zweipunktregelung geregelt.

5.7 Bewertung des autonomen Fahrzeugs

Der Ausgangspunkt des Versuches beruhte auf der Idee die Zweipunktregelung, welche die zwei Zustände Links- oder Rechtskurve mit vorgegebener Geschwindigkeit einregelt, auf ein

neuronales Netz abzubilden. Die Trainings- und Testdaten wurden mithilfe der Zweipunktregelung aufgenommen und mit diesen Datensätzen die künstliche Intelligenz angelernt. Aufgrund der Testdaten wurden somit Leistungsmaße von bis zu 95% erreicht. Für den Einsatz auf dem Arduino wurde das Netz soweit wie möglich vereinfacht und die Gleitkommazahlen der Matrizen gerundet. Trotz dieser Einschränkungen bildet der ArduRover nicht nur die Zweipunktregelung auf das neuronale Netz ab, sondern das Fahrverhalten wird sogar verbessert. Aufgrund der genannten Einschränkungen und der Tatsache, dass zum Anlernen die Fahrdaten der Zweipunktregelung verwendet wurden, ist dieser Fortschritt sehr beachtlich. Die Motoren werden jetzt nicht nur in zwei Zustände versetzt, sondern gezielt mit unterschiedlichen Geschwindigkeiten angesteuert, um der Linie optimal folgen zu können. Vor allem lange Kurven und gerade Stücke werden deutlich präzise geregelt und dadurch genauer abgefahren. Die Abbildung 5-5 zeigt die unterschiedlichen Ausgabewerte der Zweipunktregelung und des neuronalen Netzwerkes. Die dargestellten Ausgabewerte enthalten die Ansteuerungswerte des linken und des rechten Motors. Bei der Zweipunktregelung wird deutlich, dass diese nur zwei Zustände mit vorgegebenen Werten einregeln kann. Das neuronale Netzwerk kann aufgrund der Regression unterschiedliche Ausgabewerte errechnen und dadurch dynamischer auf die Gegebenheiten der Strecke reagieren.

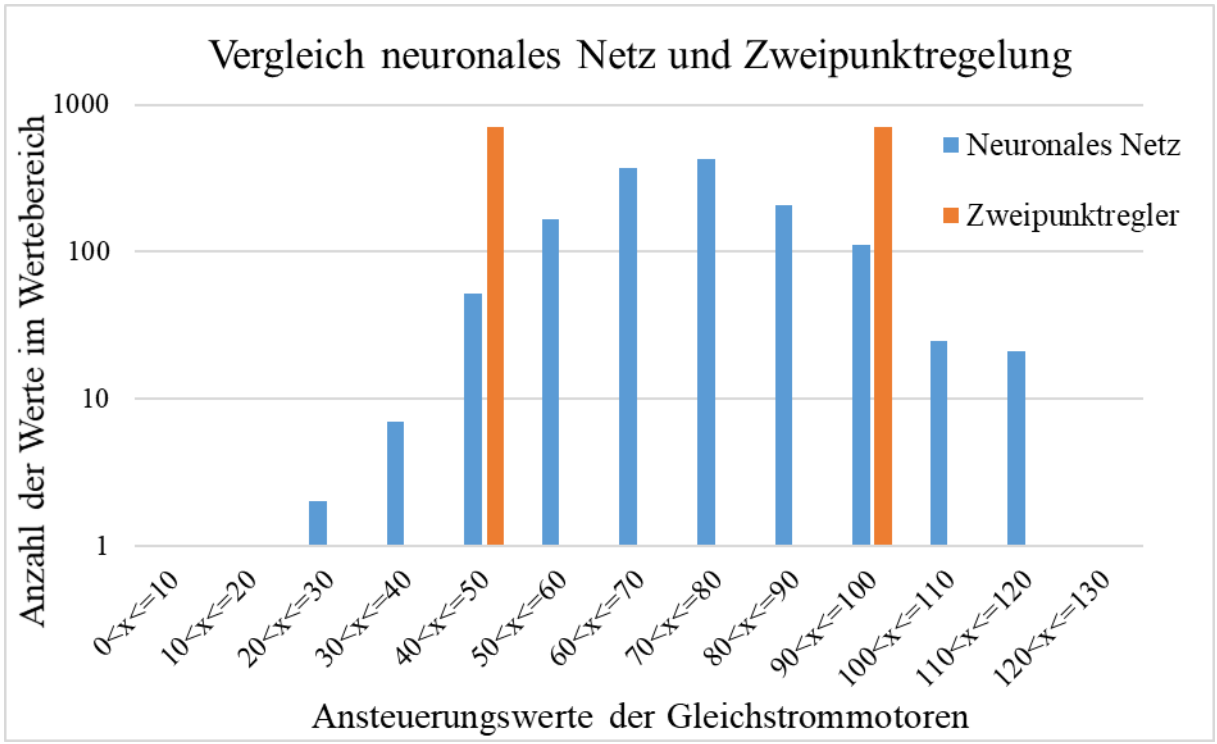

Abbildung 5-5 - Vergleich zwischen Zweipunktregelung und neuronalem Netzwerk

Die Regelung ist nicht in allen Situationen perfekt, aber fährt die vorgegebene Fahrstrecke ohne Probleme beliebig oft ab. In manchen Situationen fährt der ArduRover im Vergleich zur

Zweipunktregelung langsamer, dadurch folgt er aber der Linie besser. Auf der anderen Seite steuert das neuronale Netz bei engen Kurven den kurvenäußeren Gleichstrommotor schneller als bei der Zweipunktregelung an, damit die Kurve besser gefahren werden kann.

Die durchschnittliche Rechendauer der künstlichen Intelligenz ist mit 0,7 Millisekunden größer als die durchschnittliche Rechendauer der Zweipunktregelung mit 0,23 Millisekunden. Die längere Rechendauer hängt vor allem mit den zahlreichen Gleitkommazahlen und der Berechnung über Matrizen zusammen.

Zusammenfassend ist das neuronale Netz ein deutlicher Fortschritt im Vergleich zur Zweipunktregelung, obwohl es mit den Daten der Zweipunktregelung angelernt wurde. Dieser beachtliche Vorgang der Generalisierung funktioniert mit `TensorFlow` bereits sehr gut. Durch bessere Trainingsdaten und mehr Rechenleistung kann der schwarzen Linie nahezu ideal gefolgt werden.

5.8 Didaktische Anknüpfungspunkte

In diesem Abschnitt werden mehrere didaktische Anknüpfungspunkte für eine weitere pädagogisch-didaktische Ausarbeitung des Projekts aus dem Bereich künstliche Intelligenz gegeben.

- Wie funktionieren neuronale Netze?
- Wieso müssen neuronale Netze lernen und wie funktioniert dies?
- Lernen Menschen mit den gleichen Methoden?
- Werden Menschen irgendwann von künstlichen Intelligenzen lernen?
- Welche Aufgaben können von neuronalen Netzen übernommen werden?
- Kann ein autonomes Fahrzeug, dass zu 95% eine richtige Entscheidung trifft, im Straßenverkehr eingesetzt werden?
- Wer haftet für Unfälle, wenn eine künstliche Intelligenz eine falsche Entscheidung trifft?
- Können Trainingsdaten immer alle möglichen Szenarien erfassen und kann garantiert werden, dass ein neuronales Netz danach alle möglichen Situationen beherrscht?
- Entwickeln sich künstliche Intelligenzen irgendwann selbst weiter und übertreffen den Menschen?
- Wie weit wird der Mensch durch künstliche Intelligenz ersetzt werden können?

Kapitel 6

Zusammenfassung und Ausblick

Im Rahmen dieser Arbeit sollten die aktuellen Themengebiete künstliche Intelligenz und Datensicherheit für das Kooperationsprojekt letsgoING untersucht werden.

Das Projekt im Bereich der künstlichen Intelligenz wurde aus drei möglichen Projekten ausgewählt und mit einem neuronalen Netz realisiert. Der bereits verwendete ArduRover wurde hierzu überarbeitet und die Verfolgung einer schwarzen Linie umgesetzt. Die notwendige Software wurde erstellt und das Ergebnis validiert. Mit dem entwickelten Projekt ist es möglich, Schülern einen Einblick in die Funktionsweise künstlicher Intelligenzen zu ermöglichen. Außerdem können die Schüler selbstständig eine künstliche Intelligenz implementieren oder zumindest Teilschritte im Entwicklungsprozess nachvollziehen.

Im Bereich der Datensicherheit wurde das entwickelte Schülerprojekt auf verschiedene Schwachstellen analysiert und Angriffskonzepte entwickelt. Anschließend wurden diese Angriffskonzepte umgesetzt und ihre korrekte Funktionalität nachgewiesen. Hierbei wurde auf die Umsetzbarkeit im Schulunterricht geachtet und mögliche Lernziele erläutert. Die entstandenen Angriffskonzepte könnten beispielsweise im Rahmen eines Frontalunterrichts den Schülern vorgeführt werden.

Die technische Seite beider Themengebiete wurde im Rahmen dieser Arbeit bearbeitet. In weiteren pädagogisch-didaktische Ausarbeitungen ist es notwendig, beide Konzepte für einen Einsatz im Unterricht aufzubereiten. Hierzu müssen Ideen und Versuche entwickelt werden, wie die Schüler mit den einzelnen Komponenten experimentieren können. Des Weiteren müssen Lerninhalte und Lernziele entwickelt werden. Für die weiteren Ausarbeitungen wurden Anregungen in Form von didaktischen Anknüpfungspunkten gegeben.

Quellenverzeichnis

Literaturverzeichnis

[1] A. Badach und E. Hoffmann, Technik der IP-Netze, München: Carl Hanser Verlag , 2015.

[2] E. Bartman, Mit Arduino die elektronische Welt entdecken, Bonn: Bombini Verlags GmbH, 2017.

[3] T. Igoe, Making Things Talk, Sebastopol: O'Reilly Media, Inc., 2017.

[4] K. Schmidt, Netzwerke Grundlagen, Bodenheim : HERDT-Verlag für Bildungsmedien GmbH, 2016.

[5] P. Schnabel, Netzwerktechnik-Fibel, Ludwigsburg: Books on Demand GmbH, 2004.

[6] R. Schreiner, Computernetzwerke, München: Carl Hanser Verlag , 2016.

[7] W. Riggert, Rechnernetze, München: Carl Hanser Verlag , 2014.

[8] V. Plenk, Angewandte Netzwerktechnik kompakt, Wiesbaden: Springer Fachmedien Wiesbaden GmbH, 2017.

[9] S. Spitz, M. Pramateftakis und J. Swoboda, Kryptographie und IT-Sicherheit, Wiesbaden: Vieweg+Teubner Verlag, 2011.

[10] C. Sorge, N. Gruschka und L. Lo Iacono, Sicherheit in Kommunikationsnetzen, München: Oldenbourg Wissenschaftsverlag GmbH, 2013.

[11] G. Roden, „Botendienst," *iX*, pp. 52-56, August 2015.

[12] L. Chappel, Wireshark 101, Frechen: mitp Verlags GmbH & Co. KG, 2013.

[13] S. Dehn, Netzwerke Sicherheit, Bodenheim: HERDT-Verlag für Bildungsmedien GmbH, 2016.

[14] A. Vistola, „Kein Anschluss unter dieser URL," *Computerwoche*, pp. 18-20, 26 März 2012.

[15] P. Kroma und S. Schreiber, „heise," 15 03 2004. [Online]. Available: https://www.heise.de/security/artikel/Stoerfunk-270456.html. [Zugriff am 10 11 2017].

[16] O. von Westernhagen, „Einbruch mit Komfort," *c't*, pp. 78-83, August 2015.

[17] J. Cleve und U. Lämmel, Künstliche Intelligenz, München: Carl Hanser Verlag, 2012.

[18] W. Ertel, Grundkurs Künstliche Intelligenz, Wiesbaden: Springer Fachmedien Wiesbaden GmbH, 2016.

[19] D. Heinze, „KI im Infrarotlichtbezirk,“ *c't*, pp. 168-173, 14 Oktober 2017.

[20] C. Hadnagy, Die Kunst des Human Hacking, Frechen: mitp Verlags GmbH & Co. KG, 2011.

[21] S. Raschka, Machine Learning mit Python, Frechen: mitp Verlags GmbH & Co. KG, 2017.

[22] S. Russel und P. Norvig, Künstliche Intelligenz, München: Pearson Deutschland GmbH, 2012.

Abbildungsverzeichnis

Tabellenverzeichnis

Glossar

4

4-Wege-Handshake
Authentifizierungsvorgang in einem WLAN-Netzwerk

A

Address Resolution Protocol ARP
Übersetzt in lokalen Netzwerken die IP-Adresse in die MAC-Adresse

Agent
Programm, welches im Auftrag eines anderen Programms oder eines Menschen selbstständig Aufgaben erledigt

Aircrack
Kostenloses Programm zum Testen von WLAN Netzwerken

Aktivierungsfunktion
Simuliert das Aktivitätslevel eines Neurons

Arduino Uno
Leichtgewichtiger Mikrocontroller, der auf dem ArduRover eingesetzt wird

ArduRover
Fahrzeug mit diversen Sensoren und Aktoren

ARP-Spoofing
Methode, bei der die ARP-Cache manipuliert wird

Association
Vorgang in einem WLAN Netzwerk, bei dem jeder Netzwerkteilnehmer eine IP-Adresse zugeordnet bekommt

Authentication
Nachrichtenpaket zum Authentifizierung in einem WLAN Netzwerk

B

Botnetz
Ferngesteuertes Netzwerk von infizierten Rechnern, welches über einen Server ferngesteuert wird

Broadcast
Nachricht an alle Netzwerkteilnehmer

Broker
Bezeichnet den zentralen Server im MQTT-Protokoll

C

Cache
Lokaler temporärer Speicher

D

Deauthentication
Nachrichtenpaket zum Abmelden in einem WLAN Netzwerk

Deep Learning
Verfahren, bei dem durch Verschaltung von neuronalen Netzen die Merkmalextraktion von der KI übernommen wird

Denial-of-Serive DoS
Angriff, bei dem die Verfügbarkeit eines Dienstes eingeschränkt wird

Distributed-Denial-of-Service DDoS
Auf beliebig viele Rechner verteilter Denial-of-Service-Angriff

Domain Name System DNS
: Übersetzt die Webadresse eines Servers in die IP-Adresse und umgekehrt

Dynamic Host Configuration Protocol DHCP
: Ordnet neuen Netzwerkteilnehmern eine IP-Adresse zu

E

Einschichtiges Feedforward-Netz
: Einfachste Form eines neuronalen Netzwerkes, bei dem Eingabe- und Ausgabevektoren direkt miteinander verknüpft sind

Ettercap
: Kostenloses Programm zur Durchführung von Man-in-the-Middle-Angriffen

Exploit
: Ausnutzung einer Schwachstelle wie beispielsweise einem Implementierungsfehler

Exploit-Kits
: Programme oder Frameworks zur Generierung eines Exploits

F

Flooding
: Überschwemmen eines Netzwerkes mit Datenpaketen

Fluxion
: Kostenloses Programm für Phishing-Attacken auf WLAN Netzwerke

G

Generalisierung
: Bezeichnet im Bereich der KI die Fähigkeit, eine Logik in Datensätzen zu finden

H

Hostapd
: Kostenloses Programm, welches die Eröffnung eines Access Points ermöglicht

Hypertext Markup Language HTML
: Abstrakte Sprache zur Darstellung einer Website

Hypertext Transfer Protocol HTTP
: Protokoll zum Laden der Inhalte eines Webservers in den Webbrowser

Hypertext Transfer Protocol Secure HTTPS
: Verschlüsselte Weiterentwicklung von HTTP

I

Internet Protocol IP
: Verbindungsloses Protokoll, welches als Grundlage für die Kommunikation über Internet dient

IP-Adresse
: Logische Adresse eines Gerätes in Rechnernetzwerken, die auf dem IP-Protokoll basieren

K

Kali Linux
: Von der Firma Offensive Security entwickeltes Betriebssystem, welches auf Linux basiert und für Sicherheitstests eingesetzt wird

Klassifizierung
: Neuronales Netz bildet Eingangsparameter auf Klassen ab

Kostenfunktion
: Bestimmt in TensorFlow die Abweichung zwischen korrekten und berechneten Ergebnissen

Künstliche Intelligenz KI
: Teilgebiet der Informatik, bei dem Aufgaben möglichst effizient gelöst werden sollen und dabei die menschliche Vorgehensweise der Problemlösung nachgebildet werden soll

L

Leistungsindex
: Bewertungsmaß der KI bzw. des neuronalen Netzes

Leistungsmaß
: Bewertungsmaß der KI bzw. des neuronalen Netzes

Lernen durch Bestärkung
: Lernprozess, bei dem die Korrektheit des Eingangsvektors nicht überprüft werden kann und die KI nur in Abhängigkeit von den jeweiligen Ausgabevektoren belohnt oder bestraft wird

M

MAC-Adresse
: Eindeutige physische Geräteadresse

Man-in-the-Middle
: Angriff, bei dem der Angreifer die Kommunikation von Netzwerkteilnehmer über sich umlenkt

Maschinelles Lernen
: Teilgebiet der künstlichen Intelligenz, welches sich mit dem Lernprozess von Maschinen beschäftigt

Mehrschichtige Feedforward-Netz
: Neuronale Netzwerke mit versteckten Schichten, jedoch ohne Rückkopplung

Message Queue Telemetry Transport MQTT
: Leichtgewichtiges Protokoll für die IP-Kommunikation zwischen Geräten

Metasploit Framework
: Kostenloses Exploit-Kit

Monitormodus
: Modus des WLAN-Adapter, in dem sämtliche Informationen empfangen werden können

Mosquitto
: Kostenloses Programm für MQTT-Broker

N

Neuron
: Kleinste mathematische Einheit eines neuronalen Netzes

Neuronale Netze
: Teilbereich des maschinellen Lernens, bei dem die Struktur des menschlichen Gehirns nachgebildet werden soll

Nicht überwachtes Lernen
: Lernfahren, bei dem die KI auf Ähnlichkeiten in den Eingabevektoren reagiert

Nonce
: Zufällige einmalig verwendete Buchstaben- oder Zahlenkombination

O

Overfitting/ Überanpassung
: Die Komplexität des neuronalen Netzes ist so hoch, dass es die Trainingsdaten auswendig lernen kann und schlechte Ergebnisse bei den Testdaten erzielt

Overhead
: Zusätzliche Informationen die Datenpaketen für z.B. Flusskontrolle hinzugefügt werden

P

Pairwise Master Key PMK
: Vorher vereinbarter Schlüssel zur Authentifizierung

Pairwise Transient Key PTK
: Aus den Nonce-Werten, MAC-Adressen und Pairwise Master Key berechneter Wert

Payload
: Der Schad-Code eines Exploits

Perzeptron
: Einfachste Form eines neuronalen Netzwerkes, bei dem Eingabe- und Ausgabevektoren direkt miteinander verknüpft sind

Phishing
: Methoden des Social Engineering, um Passwörter auszuspähen

Ports
: Ermöglichen die Unterscheidung verschiedener Anwendung bei TCP und UDP

Pre Shared Key PSK
: Authentifizierungsmethode, bei der ein gemeinsames Geheimnis zur Authentifizierung genutzt wird

Publish
: Bezeichnet das Senden von Nachrichten im MQTT-Protokoll

R

Raspberry Pi
: Kompakter Einplatinencomputer

Raspian Jessie
: Kostenloses Betriebsystem für den Raspberry Pi, welches auf Linux basiert

Regression
: Neuronales Netz bildet Eingangsparameter auf Funktionswerte ab

S

Serial Peripheral Interface SPI
: Bus-System zur Übertragung von Daten

Service Set Identifier SSID
: Name eines WLAN Netzwerkes

Shell-Verbindung
: Verschlüsselte Verbindung zu einem Gerät

Slowloris
: Kostenlose Software für DoS-Angriffe auf Webserver

Social Engineering
: Beeinflussung von Menschen, um diese zu manipulieren

Subscribe
: Bezeichnet das Abonnieren eines Themas im MQTT-Protokoll

T

Teilüberwachtes Lernen
: Lernprozess, bei dem sowohl nicht überwachtes als auch überwachtes Lernen zum Training eingesetzt wird

TensorFlow
: Plattformunabhängige kostenlose Programmbibliothek für künstliche Intelligenz bzw. maschinelles Lernen

Testdaten
: Datensatz, der zur Überprüfung der Generalisierungsfähigkeit einer KI bzw. eines neuronalen Netzes verwendet wird

Trainingsdaten
: Datensatz, der zum Training einer KI bzw. eines neuronalen Netzes verwendet wird

Transmission Control Protocol TCP
: Realisiert verbindungsorientierte Kommunikation basierend auf dem Internet Protocol

U

Überwachtes Lernen
: Lernverfahren, bei dem ein Lehrer zu jedem Eingangsvektor den korrekten Ausgangsvektor zur Verfügung stellt

URL
: Webadresse zur Erreichung von Webinhalten

User Datagram Protcol UDP
: Realisiert verbindungslose Kommunikation basierend auf dem Internet Protocol

V

Versteckte Schichten
: Bezeichnet die Schichten zwischen der Ein- und der Ausgabeschicht eines neuronalen Netzes

W

Webserver
: Dienst, der Webinhalte unter bestimmten Webadresse bereit hält

WiFi-Protected-Acces WPA
: Aktueller Standard zur Verschlüsselung von WLAN-Netzwerken

Wired Equivalent Privacy WEP
: Entschlüsselter Standard zur Verschlüsselung von WLAN Netzwerken

Wireless Local Area Network WLAN
: Drahtloses lokales Funknetzwerk

Wireshark
: Kostenloses Programm zur Analyse von Datenprotokollen

Wörterbuch-Attacke
: Angriff, bei dem ein Passwort durch Ausprobieren herausgefunden wird

Eidesstattliche Erklärung

Ich versichere, dass ich diese Arbeit ohne fremde Hilfe selbstständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie alle wörtlichen oder sinngemäß übernommenen Stellen in der Arbeit gekennzeichnet habe. Die Arbeit wurde noch keiner Kommission zur Prüfung vorgelegt und verletzt in keiner Weise Rechte Dritter.

Reutlingen, 12.01.2018 D. Fuchs

Ort, Datum Dennis Fuchs