Daniel Pantjuskin-Moos
108013248222
Vincent König
108011232630
Gruppe: D

Abgabe PHYSEC 5

1 Implementierung, Analyse, Theorie

Der *Sourcecode* und die *Plots* können hier heruntergeladen werden.

1.1 Analyse I

- Vergleichen Sie das obige Messsetup mit dem von Ihnen verwendeten Setup während des Praktikums und setzen Sie die mögliche Menge an extrahierbaren Bits in Vergleich zu den RSSI-Werten

Da im jetzigen Assignment 5 mit CSI-Daten gearbeitet wird, ist die zu verarbeitende Menge an Daten wesentlich größer.

- Überlegen Sie, welchen Vorteil RSSI-Werte gegenüber den komplexen Kanalimpulsantworten haben

RSSI-Werte sind einfacher zu verarbeiten, da sie weniger Rechenaufwand nach sich ziehen und weniger Speicherplatz benötigen. Außerdem ist die Programmierarbeit einfacher für RSSI-Werte.

1.2 Theorie

Hierbei wurde sich unter anderem auf dieses Paper (Seite 1132) und diese Doktorarbeit (Seite 20 ff.) gestützt, welche kostenlos downloadbar sind.

Intradistanz

Beschreibt die Distanz zwischen den *Responses* für die selbe *Challenge*. Man kann es als Indikator für die Größe des Messfehlers, als auch laut dem Paper als Messwert für die Reproduzierbarkeit betrachten. Die Unterschiede entstehen durch Umwelteinflüsse und *statistical Noise*.

In der Praxis kann dieser Wert, ähnlich wie in den Kommentaren zu `compute_intra_distance` von Aufgabe 1.3 beschrieben, berechnet werden: Die erste Messung einer *Challenge* wird als Referenz genommen. Anschließend wird der euklidsche Abstand zu allen Messungen der selben *Challenge* berechnet. Das wird für alle *Challenges* wiederholt.

Interdistanz

Distanz zwischen *Responses* für verschiedene *Challenges* und laut dem Paper ein Indikator für die Einzigartigkeit eines *PUF circuit* beziehungsweise laut Vorlesung 5, Folie 36 ein Indikator für die Einzigartigkeit der *responses*.

In der Praxis, kann es ähnlich wie in der Aufgabe 1.3 bei der Funktion `compute_inter_distance` aussehen, wobei man hier die Anzahl der zu berechnenden Distanzen gering halten möchte. Deswegen beschränkt man sich auf die jeweils ersten Messungen jeder *Challenge*: Dazu werden alle Distanzen aller Paare, ungeordnet und ohne Zurücklegen, gebildet von den ersten Messungen aller *Challenges*, was somit $\binom{40}{2} = 780$ verschiedene Paare sind.

1.3 Implementierung

```
def standardize_per_channel(data):
  print("standardize_per_channel...")
  data_copy = data.copy()
  new_data = []

  for measurement in data_copy:
    #get channels
    channels=np.array(np.split(measurement, len(measurement)/114))
    # copied new data in 1D Array
    new_data.append([ ((chan-mean)/var).flatten() for chan,mean,var in zip(
                                        channels, channels.mean(axis=1),
                                        channels.var(axis=1)) ])

  print('sucessful.')
  return np.array(new_data)
```

```
def euclidean_norm_per_channel(data):
  print("euclidean_norm_per_channel...")
  data_copy = data.copy()
  new_data = []

  # len of vector
  l = lambda vector_a: compute_euclidean_distance(vector_a, np.zeros_like(
                                        vector_a))

  for measurement in data_copy:
    channels=np.array(np.split(measurement, len(measurement)/114))
    new_data.append([ (chan/l(chan)).flatten() for chan in channels ])


  print('sucessful.')
  return np.array(new_data)
```

```
def compute_intra_distance(data, num_challenges):
  print("compute_intra_distance...")
  data_copy = data.copy()

  challenges = np.split(data_copy, num_challenges)
  intra_distances = [compute_euclidean_distance(challenge[0],measurement)
                       for challenge in challenges
                       for measurement in challenge[1:]]

  print('sucessful.')
  return np.array(intra_distances)
```

```
def compute_inter_distance(data, num_challenges):
  print("compute_inter_distance...")
  data_copy = data.copy()

  challenges = np.split(data_copy, num_challenges)
  inter_distances = [compute_euclidean_distance(chall[0],measurement)
                       for i,chall in enumerate(challenges)
                       for j,subchall in enumerate(challenges)
                       if subchall is not chall
                       for measurement in subchall[0 if i<j else 1:]]

  print('sucessful.')
  return np.array(inter_distances)
```

1.4 Analyse II

- Ist eine Überlappung zwischen Intra- und Interdistanzen gut oder schlecht?

Laut Folie 36 der fünften Vorlesung sollte die Intradistanzen deutlich kleiner sein als die Interdistanzen, ansonsten hat man kein funktionierendes System: „Measurement error must always be low enough to distinguish challenges (intra distance « inter distance). Otherwise, the system cannot be used!“

- Welche Bedeutung spielt die Größe der Überlappung im Bezug auf die Realisierung einer strong-PUF?

Die Überlappung sollte gering, sein wenn man eine strong-PUF konstruieren möchte. Wie in der Vorlesung erwähnt wurde zur Folie 36 kann man verschiedene *Responses* und *Challenges* nicht außereinander halten, wenn die Intradistanz groß ist. Eine klare Einzigartigkeit der *Responses* ist erwünscht.

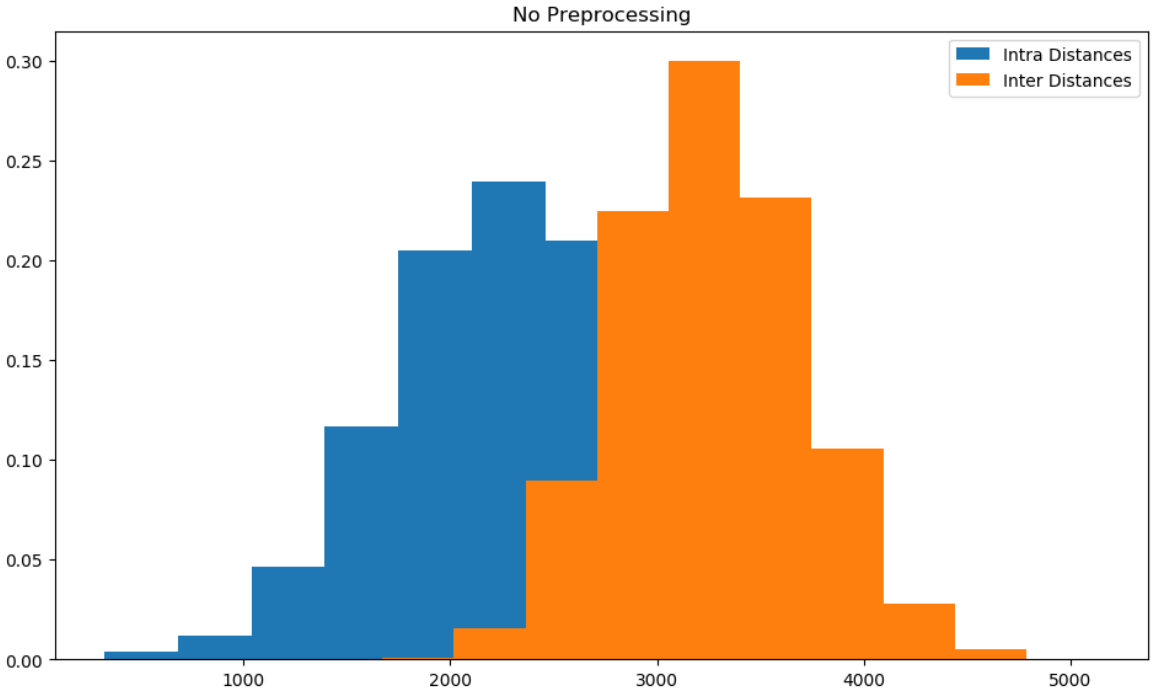
- Treffen Sie zu den verschiedenen Plots des Frameworks Aussagen, wie leicht bzw. schwer es ein Angreifer hat, eine Challenge richtig zu raten, wenn er nur die Daten anderer Challenges besitzt. Welche Rolle spielen dabei die verschiedenen Arten des Preprocessings?

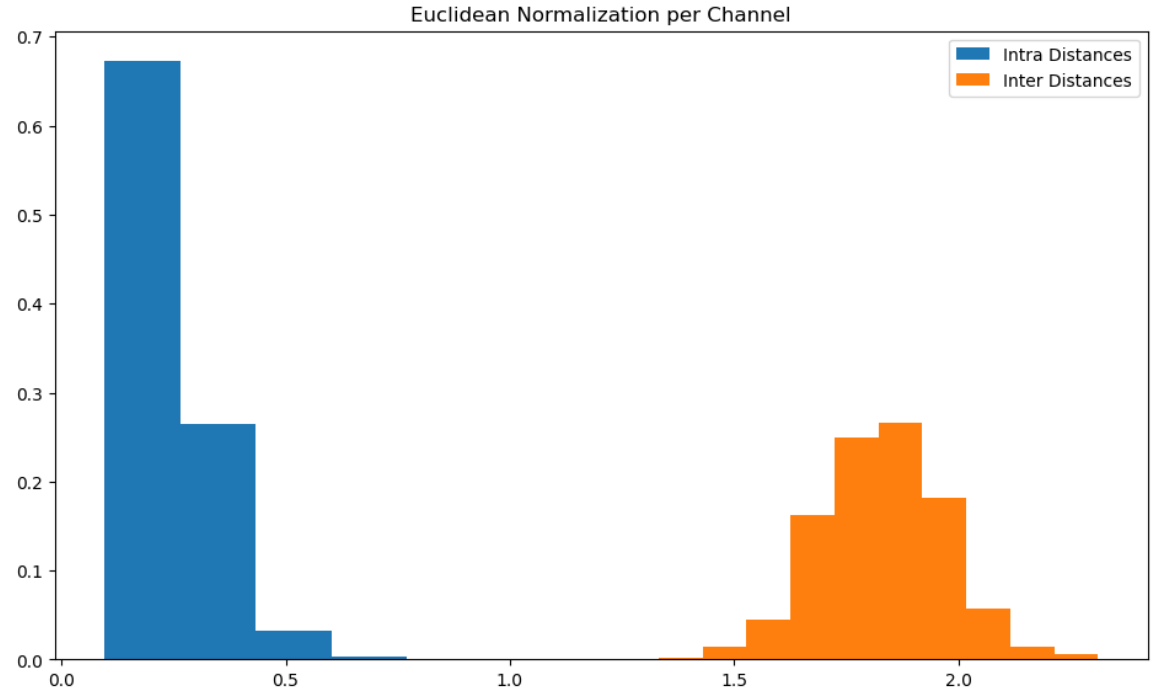
`No Preprocessing`: Überlappungs ist hier signifikant. Ein Angreifer hat hier mit Abstand die besten Möglichkeiten erfolgreich zu sein.

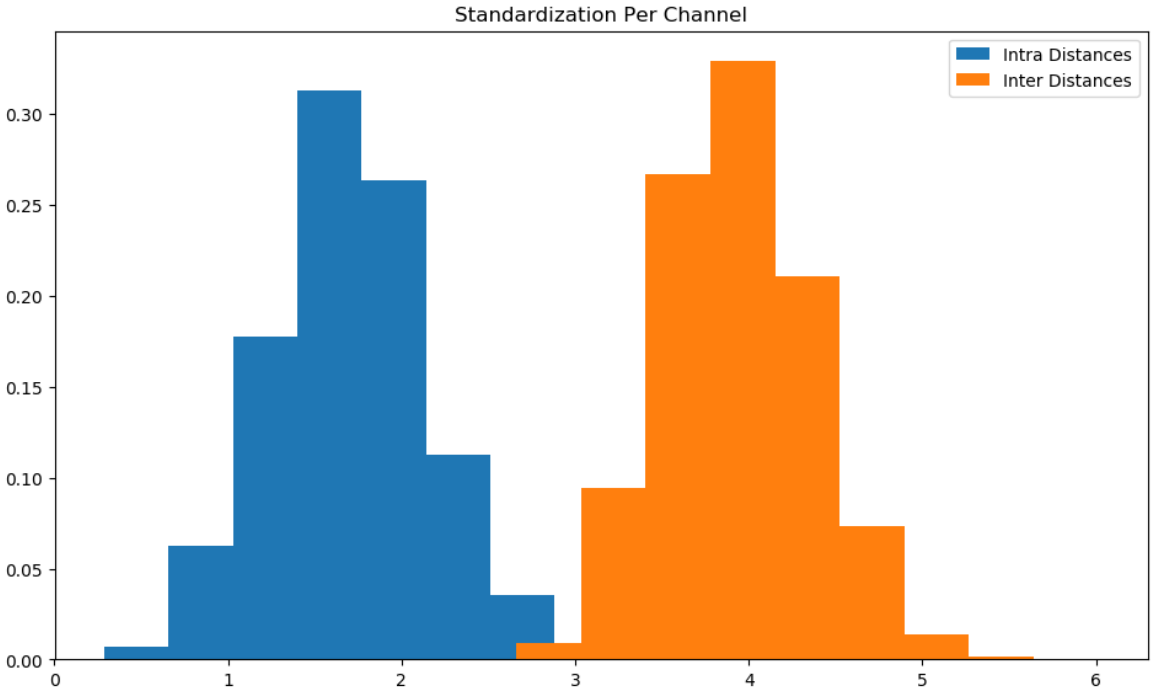
`Euclidean Normalization per Channel`: Intradistanz ist gering und fällt streng monoton ab. Die Verteilung der Interdistanzen hingegen ähnelt einem Hügel mit der Mitte um den Wert von etwa 1.8 herum und zu beiden Seiten streng monoton abfallend. Der Abstand zwischen den *Peaks* beider Verteilungen ist bisschen weniger als 1,8. Die Erfolgsaussichten eines Angreifers sind hier wesentlich geringer als oben.

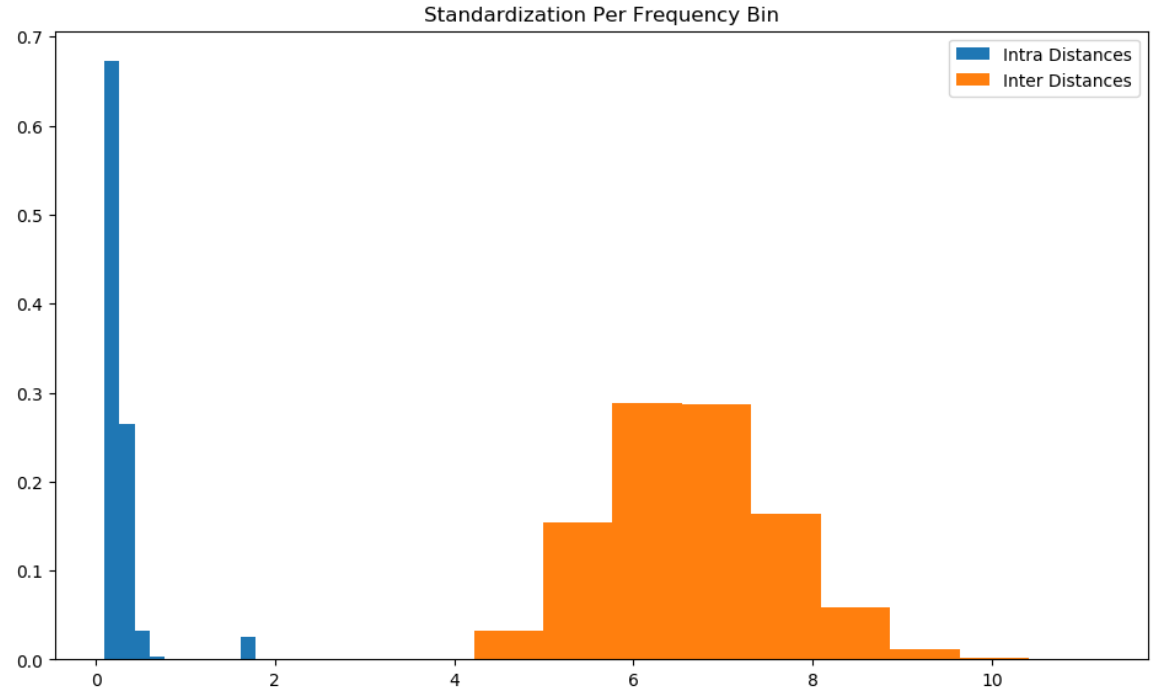
`Standardization per Channel`: Hier gibt ein wenig Überschneidung. Dieses *Preprocessing* bietet im Vergleich zu den anderen *Preprocessings* einem Angreifer die besten Chancen.

`Standardization per Frequency Bin`: Optisch ähnelt die Verteilung der euklidschen Normalisation pro Kanal. Auffallend ist, dass die Verteilung bei Intradistanz nicht streng monoton abfallend ist. Der Verteilung bei Interdistanz erstreckt sich über fast eine Spannweite von 6. Hier hat der Angreifer die schlechtesten Chancen.


No Preprocessing
Intra Distances
Inter Distances
0.30
0.25
0.20
0.15
0.10
0.05
0.00
1000
2000
3000
4000
5000


Euclidean Normalization per Channel
Intra Distances
Inter Distances
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0.0
0.5
1.0
1.5
2.0


Standardization Per Channel
Intra Distances
Inter Distances
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
1
2
3
4
5
6


Standardization Per Frequency Bin
Intra Distances
Inter Distances
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
2
4
6
8
10

2 Reading Assignment

- Wie groß muss die effektive Bandbreite beim *Indoor Positioning* sein um 1cm genaue Positionen bestimmen zu können? Was ist das Problem mit handelsüblichen WLAN APs?

  „When the effective bandwidth reaches 360 MHz, the region of ambiguity shrinks to a ball with an approximately 1-cm radius, which indicates centimeter-level accuracy. Unfortunately, the bandwidths on mainstream 802.11n Wi-Fi chips are merely 20 or 40 MHz, insufficient for centimeter-level indoor positioning."
  Die effektive Bandbreite muss also 360 Mhz betragen um diese Genauigkeit erreichen zu können, jedoch haben gewöhnliche Wi-Fi Chips nur eine Bandbreite von 20 bis 40 MHz.

- Welche Phasen müssen durchlaufen werden, damit *Wireless Event Detection* stattfinden kann? Beschreiben Sie diese kurz.

  Phase 1: Offline training
  Für jedes *Indoor Event* wird die dazugehörige CSI mithilfe von *Channel Probing* erlangt. Daraus wird eine Matrix gebildet.

  Phase 2: Online testing
  Das Ziel eines TRIEDS ist das Auftreten jedes *Training Indoor Events* zu ermitteln und zwar mithilfe der Auswertung der Ähnlichkeit der CSI im Testlauf und im Trainingslauf. Letztere sind in der Trainings- Datenbank $\mathcal{G}$ gespeichert. Die unverarbeiteten CSI-Daten, welche durch Radiogeräte gewonnen wurden, sind komplex-wertig und stark multidimensional, was die Verarbeitung deutlich erschwert.

- Welche physikalischen Eigenschaften eines Menschen beeinflussen unter anderem die Human Radio Biometrics und weshalb sind sie ungeeignet als Messwerte?

  „According to the literature, the wireless propagation pattern around a human body depends highly on individual physical characteristics (e.g., height and mass), the total body water volume, the skin condition, and the characteristics of other biological tissues"

  Physikalische Eigenschaften des Körpers wie Körpergröße, Gewicht, Wasseranteil und Zustand der Haut beeinflussen *Human Radio Biometrics*.

  „However, the human body may affect only a few paths of the multipath CSI, and the energy of those paths is small because of the low reflectivity and permittivity compared with other static objects, such as the walls and furniture. As a result, human radio biometrics captured through radio shot are buried in the CSI by other useless components."

  Es stellt sich heraus, dass aufgrund der geringen Reflexivität und Permeabilität des menschlichen Körpers im Vergleich zu anderen statischen Objekten wie Wänden und Möbeln, die anvisierten CSI-Pfade eine geringe Energie haben. Das hat zur Folge, dass durch Radio aufgenommene *Human Radio Biometrics* von anderen, nutzlosen Komponenten (größtenteils) übertüncht werden.

- Erläutern sie in eigenen Worten, wie die beiden erwähnten Methoden des Indoor Trackings funktionieren.

  Triangulation
  Bei dieser Methode wird versucht entweder die Distanz oder den Winkel zwischen dem Gerät und mehreren *Anchors* zu schätzen. Auf die Position des Geräts kann durch geometrische Triangulation geschlussfolgert werden. Die Distanz kann abgeschätzt werden, wenn man die empfangenen Pakete hinsichtlich RSSI-Abstieg oder Sendedauer analysiert. Der Winkel zwischen den Geräten kann extrahiert werden, wenn man die Eigenschaften der CSI-Werte, die von mehreren Antennen empfangen wurden, untersucht.

  Fingerprinting
  Bei dieser Methode können die benötigten Eigenschaften entweder von detaillierten CSI-Werten einer bestimmten Position zu allen *Anchors* in Reichweite oder aus den RSSI-Vektoren entnommen werden. Ein Nachteil ist, dass die *Fingerprint*-Datenbank die zugeordnete *Fingerprints* sammelt vor Wiederbenutzen aktualisiert werden muss, was eine Folge der Empfindlichkeit auf Umwelteinflüsse ist. Außerdem ist der Rechenaufwand groß und somit auch die Latenz.