JUSTUS-LIEBIG-
UNIVERSITÄT
GIESSEN

Bachelorthesis

Identifizierung von langsamen Pionen durch Support Vector Machines

Machine Learning-basierte Datenauswertung für das Belle II Experiment

Timo Schellhaas

II. Physikalisches Institut
Experimental Physics II

Belle II

Justus-Liebig-Universität Gießen
II.Physikalisches Institut
Fachbereich 07

Bachelorthesis

Identifizierung von langsamen Pionen durch Support Vector Machines

Identifying Slow Pions using Support Vector Machines

Timo Schellhaas
Gießen, 25.02.2022

Betreuer: Apl. Prof. Dr. Jens Sören Lange
Zweitkorrektur: Prof. Dr. Claudia Höhne

Inhaltsverzeichnis

Zusammenfassung/ Abstract

Im November 2021 veröffentlichte Johannes Bilk seine Masterthesis über die Klassifizierung langsamer Pionen am Belle II Experiment. Ziel der Arbeit war es, ein Deep Learning Modell zu erstellen, welches langsame Pionen in den innersten Schichten des Detektors identifiziert, da diese wegen ihres geringen Impulses nicht die äußeren Schichten des Detektors erreichen können. Hierfür verwendete er neuronale Netze und kam auf einen Genauigkeit von bis zu 83%, je nach Modell. Aufbauend auf dieser Arbeit, wurde für diese Bachelorthesis versucht, die langsamen Pionen über Support Vektor Maschinen zu identifizieren und möglicherweise besser Ergebnisse zu erzielen.

In November 2021 Johannes Bilk published his masters' thesis about the classification of slow pions at the Belle II experiment. The goal of that thesis was to implement a deep learning model, which could identify the slow pions in the most inner layers of the detector, because the slow pions can't reach the outer layers due to their low momenta. He used neural networks and achieved accuracies up to 83%, depending on the model he used. Consequtively to his work, this bachelors' thesis aims to identify slow pions using support vector machines and, if possible, improve his results.

1 Einleitung

Mit dem Standardmodell der Teilchenphysik wurde ein Grundstein für das Verständnis subatomarer Materie gelegt. Bisher lassen sich jedoch nicht alle bekannten Prozesse allein auf dieses zurückführen, weshalb versucht wird, das Modell so zu erweitern, oder gar zu ersetzen, dass diese Prozesse hinreichend beschrieben werden können. [1]

Pionen mit einem geringen Impuls, sogenannte langsame Pionen (englisch: slow pions) sind in Zerfallsprozessen involviert, die möglicherweise neue Erkenntnisse für das Standardmodell liefern. Deshalb werden solche langsamen Pionen beim KEK in Japan im Belle II Experiment untersucht. [2]

Durch ihren geringen Impuls bzw. ihrer geringen Geschwindigkeit ist es jedoch sehr schwer langsame Pionen zu detektieren, da diese fast ausschließlich nur den inneren Detektor des Belle II erreichen, jedoch nicht die äußeren. Um dieses Problem zu lösen, wurde vorgeschlagen, ein künstliches Intelligenz Modell zu entwickeln, welches die Detektionsrate erhöhen soll, indem die räumliche Orientierungen der langsamen Pionen im inneren Detektor getrackt werden.

Dafür kommen prinzipiell mehrere Modelle in Fragen, die miteinander verglichen werden können. In dieser Bachelorthesis werden die Ergebnisse von Support Vector Machines vorgestellt und mit den Ergebnisse der Masterthesis von Johannes Bilk (Justus - Liebig Universität, 2021)[3], welcher Neuronale Netze verwendet hat, verglichen.

2 Belle II Experiment

Das Belle II Experiment am Teilchenbeschleuniger SuperKEKB ist der direkte Nachfolger des Belle Experiments am KEKB. Das Ziel sowohl von Belle als auch Belle II war bzw. ist die Untersuchung von B-Mesonen, welche unter anderem von den späteren Nobelpreisträgern Makoto Kobayashi und Toshihide Maskawa vorhergesagten Effekt der CP-Verletzung bestätigen konnten. Diese könnte eine entscheidende Rolle bei der Erklärung der Dominanz von Materie gegenüber Anti-Materie spielen. [4]

2.1 SuperKEKB

Der asymmetrische Elektron-Positron Teilchenbeschleuniger SuperKEKB befindet sich in Tsukuba, Japan. Abbildung 2.1 zeigt dabei den schematischen Aufbau. SuperKEKB besitz zwei Ringe, einen für den Elektronen und einen für den Positronenstrahl, mit einem Umfang von jeweils ca. 3 km. Dabei erreichen die Elektronen eine Energie von 7 GeV bei einem Strahlstrom (englisch: "beam current") von $2,6$ A und die Positronen eine Energie von 4 GeV bei einem Strahlstrom von $3,6$ A. Dies erzeugt eine Schwerpunktenergie (englisch: center-of-mass energy) $10,58$ GeV. Zudem erreicht der SuperKEKB eine Luminosität von $6,5 \cdot 10^{35}$ $\frac{1}{cm^2 s}$, welche die Begegnungen von Teilchen pro Zeit pro Fläche beschreibt und somit eine Kenngröße der Leistungsfähigkeit eines Teilcheneschleunigers ist.

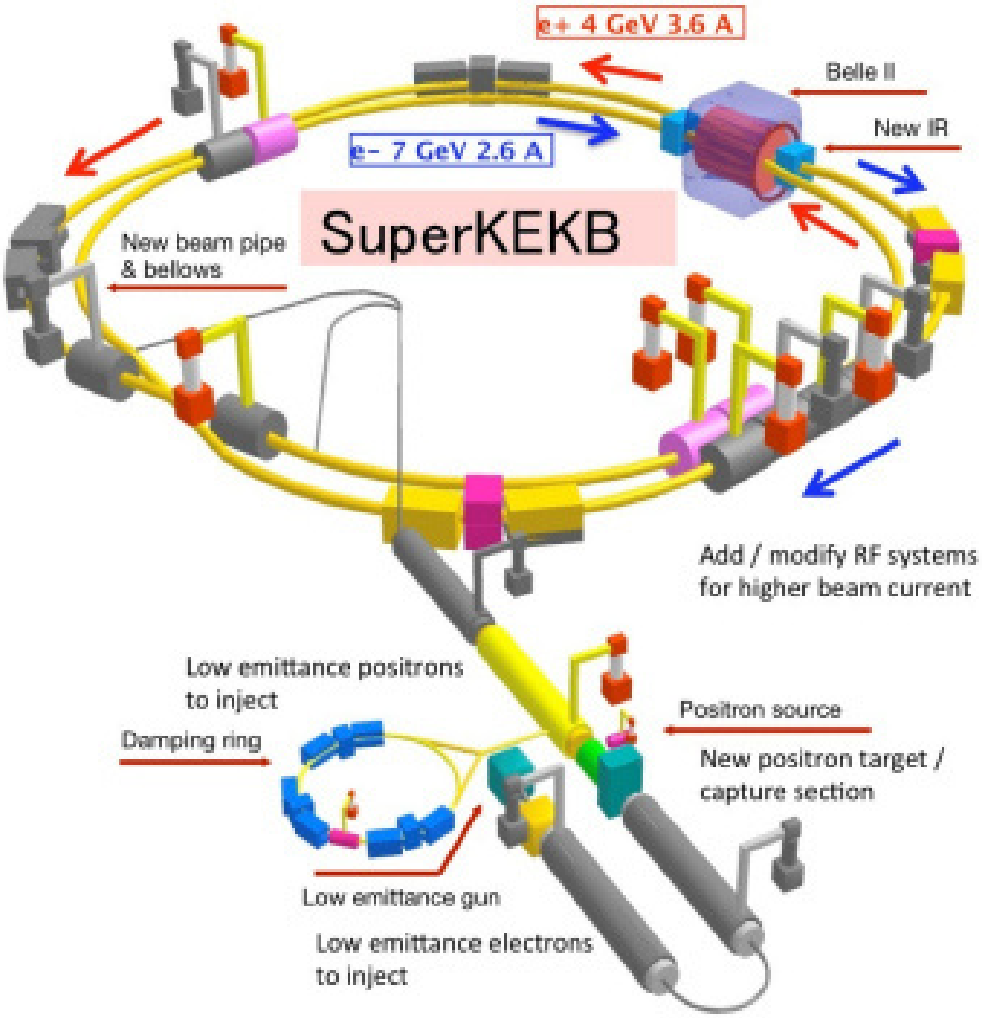
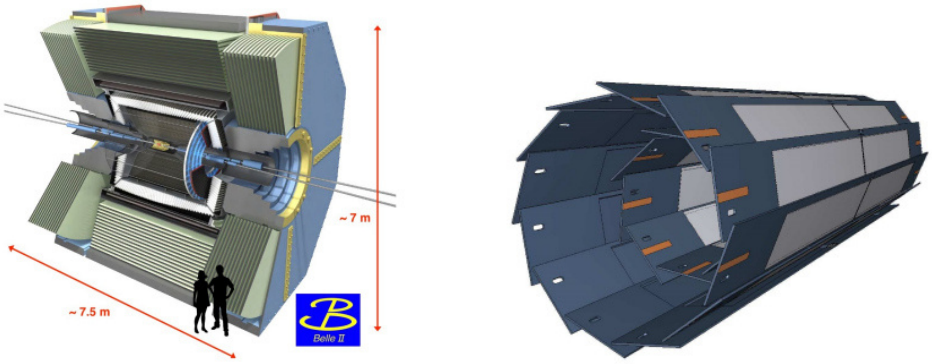


Abb. 2.1: Beschleunigeranlage SuperKEKB [5]

2.1.1 Subdetektoren

Abbildung 2.2 zeigt den Aufbau des Belle II Detektors. Dieser besteht aus mehreren Subdetektoren, welche im Folgenden näher erläutert werden [3].



(a) (b)

Abb. 2.2: Aufbau des Belle II Detektors. a) zeigt den Querschnitt des gesamten Detektors, b) zeigt den PXD [3], [5]

PXD:
Der PXD (pixel detector) ist der innere Subdetektor von Belle II und besteht aus zwei Silizium-Schichten (englisch: layers), wobei diese aus insgesamt 200,000 DEPFET-Pixelzellen (englisch: depleted p-channel field effect transistor, deutsch: Feldeffekttransistor mit verarmtem p-Kanal) bestehen, die sich auf eine Fläche von insgesamt 8 cm^2 verteilen). Diese beiden Schichten haben die Radien 14 mm und 22 mm, sowie insgesamt 10^6 Auslesekanäle (englisch: readout channels).

SVD:
Der SVD (silicon vertex detector) besteht aus vier doppelseitigen Silizium Schichten mit Radien von 39 mm, 88 mm, 104 mm und 135 mm und insgesamt $224 \cdot 10^3$ Auslesekanälen. Durch den SVD können die Vertex Positionen der erzeugten Teilchen rekonstruiert werden.

CDC:
Die CDC (central drift chamber) misst die Trajektorien, den Impuls und den Energieverlust $\frac{dE}{dx}$ der erzeugten Teilchen. Dabei besitzt sie einen inneren Radius von 160 mm und einen äußeren Radius von 1130 mm. Zudem ist die Kammer mit $He-C_2H_6$ Gas gefüllt.

TOP Der TOP (time-of-propagation) ist ein Detektor, welcher aus 16 Quarz Stangen ($260\ cm \times 45 cm \times 2\ cm$) besteht und dafür konzipiert wurde, geladene Teilchen zu identifizieren und Kaonen von Pionen zu trennen.

ARICH Der ARICH (aerogel ring-imaging cherenkov detector) ist ebenso wie der TOP für die Identifizierung von geladenen Teilchen bzw. der Trennung von Kaonen und Pionen zuständig, jedoch ist der ARICH im Gegensatz zum TOP nicht konzentrisch um die Strahlrichtung gebaut, sondern steht senkrecht zu dieser.

ECL Das ECL (electromagnatic calorimeter) ist etwa 3 m lang und besitzt einen inneren Radius von $1,25\ m$ bei einem Gewicht von ca. 43 t. Durch das ECL werden hauptsächlich Pionen von Elektronen getrennt.

KLM Das KLM (K_L^0 and muon detector) wird zur Messung von hadronischen Showern benutzt.

Von größerer Bedeutung für diese Arbeit sind vor Allem der PXD und der SVD. Das liegt daran, dass es das Ziel ist, möglichst viele Pionen mit dem SVD detektieren zu können.

2.2 Theorie

Um besser zu verstehen, warum die Untersuchung von langsamen Pionen wichtig ist, muss zunächst der physikalische Hintergrund beleuchtet werden. Da es das Ziel der Untersuchung ist, das Standardmodell zu erweitern, wird dieses zunächst grob zusammengefasst, bevor die langsamen Pionen und die damit zusammenhängenden Effekte thematisiert werden.

2.2.1 Standardmodell der Teilchenphysik

Durch das Standardmodell der Teilchenphysik werden die (uns heute bekannten) elementaren Teilchen und Wechselwirkungen zusammengefasst, wobei hier die Quantenfeldtheorie die Grundlage bildet. Das Standardmodell ist in Abbildung 2.3 dargestellt.

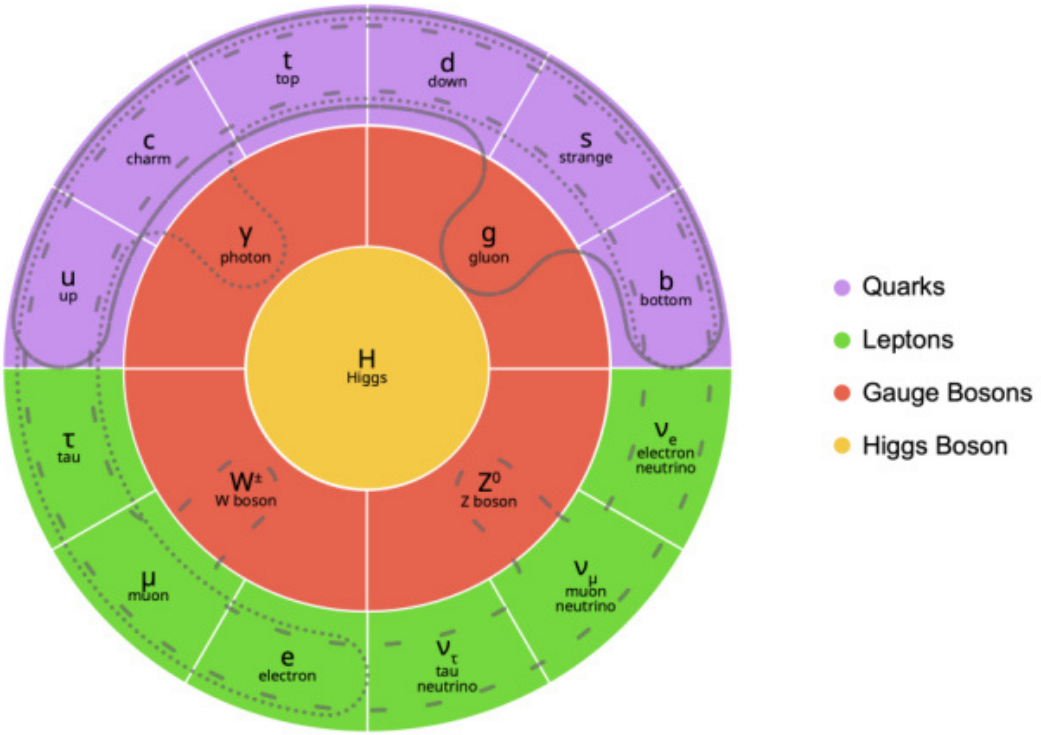


Abb. 2.3: Standardmodell der Teilchenphysik

Das Standardmodell besteht aus mehreren verschiedenen Teilchenklassen und Wechselwirkungen. Auf diese Klassifizierung und weitere wichtige resultierende Teilchen wird in der folgenden Aufzählung genauer eingegangen: [6]

1. *Fermionen*:
   Fermionen sind Teilchen, die einen halbzahligen Spin besitzen und, anschaulich gesprochen, diejenigen Teilchen, aus denen Materie besteht. Der Spin ist dabei der Eigendrehimpuls eines Teilchens und ebenso wie die Masse eine unveränderbare Eigenschaft eines Teilchens. Im Standardmodell liegen zwei verschiedene Klassen von Fermionen als Elementarteilchen vor: Die Quarks und die Leptonen, welche jeweils in drei Generationen unterteilt sind.

2. *Quarks:*
   Es existieren insgesamt sechs Quarks und deren Antiquarks. Diese werden be-

zeichnet als u (up), d (down), c (charm), s (strange), t (top) und b (bottom). u, c und t besitzen eine elektrische Ladung von $\frac{2}{3}e$, d, s und b eine elektrische Ladung von $-\frac{1}{3}e$, bzw. deren Antiteilchen dieselbe elektrische Ladung mit dem Faktor -1 multipliziert. Dabei können die Zusammensetzungen der Quarks, welche Hadronen genannt werden, einen ganzzahligen Spin (Mesonen, Quark-Antiquark Paar) oder einen halbzahligen Spin (Baryonen, drei Quarks) besitzen. Es können auch exotische Hadronen auftreten, die in dieser Arbeit allerdings nicht weiter behandelt werden. Zudem besitzen alle Quarks eine Masse die größer als 0 ist.

Quarks tragen zudem eine Farbladung rot, grün oder blau (bzw. antirot, antigrün oder antiblau). Diese ist veränderbar und Grund für die Teilnahme an der starken Wechselwirkung. Gemäß der Quantenchromodynamik muss die Zusammensetzung von Quarks farblos sein, also müssen alle drei Farben beziehungsweise alle drei Antifarben vorkommen (Baryonen) oder eine Farbe und seine Antifarbe (Mesonen). Aus diesem Grund ist es nicht möglich, dass Quarks isoliert vorkommen. Zudem nehmen die Quarks, beziehungsweise die Hadronen an der schwachen und der elektrischen Wechselwirkung teil.

3. *Leptonen:*
   Ebenso wie die Quarks, existieren auch sechs Leptonen und deren jeweiligen Antiteilchen. Diese sind e (Elektron) μ (Myon) und τ, sowie die dazugehörigen Neutrinos ν_e, ν_μ und ν_τ. Diese tragen die elektrische Ladung $-1e$ bzw. die Neutrions die elektrische Ladung 0. Die Leptonen nehmen nicht an der starken Wechselwirkung teil, da sie keine Farbladung besitzen. Neutrinos nehmen nicht an der elektrischen Wechselwirkung teil, da sie keine elektrische Ladung tragen. Jedoch nehmen alle Leptonen an der schwachen Wechselwirkung teil.

4. *Bosonen:*
   Bosonen besitzen im Gegensatz zu den Fermionen einen ganzzahligen Spin. Die elementaren Bosonen sind für Wechselwirkung zwischen den Teilchen verantwortlich, weshalb man auch von Austauschteilchen spricht. Es existieren sechs solcher Bosonen: γ (Photon), g (Gluon), Z^0 (Z-Boson), $W^{+/-}$ (W+ und W- Bosonen) und H (Higgs-Boson). Dabei ist γ das Austauschteilchen

der elektromagnetischen Wechselwirkung, g das Austauschteilchen der starken Wechselwirkung und $Z^0, W^{+/-}$ die Austauschteilchen der schwachen Wechselwirkung. Das Higgs-Boson nimmt dabei einen besonderen Platz ein, da seine Wechselwirkung den Teilchen ihre Masse verleiht, jedoch nicht explizit Austauschteilchen einer der vier Elementarkräfte ist. Das (noch hypothetische) Austauschteilchen der gravitativen Wechselwirkung konnte bislang nicht in das Standardmodell integriert werden.

Es bleibt anzumerken, dass das Standardmodell keineswegs den Anspruch auf Vollständigkeit erhebt. Ganz im Gegenteil: es werden Effekte und Phänomene in der Natur beobachtet, welche nicht durch das Standardmodell zu beschreiben sind. Deshalb wird versucht das Standardmodell zu erweitern. [6]

2.2.2 CP Verletzung

In der Teilchenphysik gibt es eine Vielzahl von Erhaltungssätzen, die durch die entsprechenden Wechselwirkung festgelegt werden. Ist ein Erhaltungssatz verletzt, so kann ein physikalischer Prozess, zum Beispiel der Zerfall eines Teilchens in zwei andere Teilchen, nicht stattfinden. [6]

Ein wichtiger Erhaltungssatz ist die CP-Erhaltung, wobei das C für Charge (deutsch: Ladung) und das P für Parity (deutsch: Parität) steht, durch die schwache Wechselwirkung nicht erhalten ist. Man spricht dabei auch von CP-Verletzung.

Historisch wurde zunächst durch das Wu-Experiment im Jahr 1956 die Paritätsverletzung festgestellt und ein Jahr später durch das Goldhaber-Experiment, dass es nur linkshändige Neutrinos bzw. nur rechtshändige Antineutrinos gibt. [7]

Zunächst nahm man noch an, dass trotz der bewiesenen P-Verletzung die C-Parität auch durch die schwache Wechselwirkung erhalten bleibt, jedoch wurde diese Annahme 1964 durch den Zerfall von neutralen Kaonen von James Cronin und Val Fitsch widerlegt. [8]

Dabei wurde bei dem Zerfall von K_L^0-Mesonen erwartet, dass diese in drei Pionen zerfallen und von K_S^0, dass diese in zwei Pionen zerfallen. Tatsächlich wurde aber

festgestellt, dass K_L^0 zu einem kleinen Anteil (etwa 0.2%) in zwei Pionen zerfallen:

$K_L^0 \rightarrow \pi^+\pi^-\pi^0$ (CP=-1)
$K_L^0 \rightarrow \pi^+\pi^-$ (CP=+1)

Durch diese Erkenntnis konnte nun fortan angenommen werden, dass die schwache Wechselwirkung die Ladungsparität ändern kann. [9]

2.2.3 B-Mesonen

Um die CP-Verletzung weiter zu untersuchen, bieten B-Mesonen interessante Zerfallsprozesse, weshalb diese beim Belle II Experiment durch die Kollision von Elektronen und Positronen erzeugt werden. Dabei betrachtet diese Arbeit insbesondere den Zerfall von B^0 und $\bar{B}^0$ Mesonen, da diese in D-Mesonen zerfallen, welche wiederum in langsame Pionen zerfallen:

$B^0 \rightarrow D^{*-} + X^+$
$D^{*-} \rightarrow \pi^-$

$\bar{B}^0 \rightarrow D^{*+} + X^-$
$D^{*+} \rightarrow \pi^+$

Dabei stehen die $X^\pm$ für beliebige Mesonen oder geladene Leptonen mit dazugehörigen Neutrinos. Die Pionen die dabei erzeugt werden, werden als langsame Pionen bezeichnet[2][10].

2.2.4 Langsame Pionen

Langsame Pionen sind, wie es der Name vermuten lässt, Pionen, welche einen geringen Impuls tragen. Pionen sind Mesonen die aus einem $u\bar{d}$ (π^+), einen $\bar{u}d$ (π^-) Paar bestehen, oder aus einem quantenmechanischen Mischzustand $\frac{1}{\sqrt{2}}[|u\bar{u}> - |d\bar{d}>]$ (π^0).[10][11]

Ziel bei der Detektion der langsamen Pionen ist es, die B-Mesonen rekonstruieren

zu können. Das Problem liegt allerdings darin, dass die Pionen, durch den geringen Impuls, nicht über die inneren Schichten der Detektoranlage hinaus detektiert werden können. Abbildung 2.4 zeigt diesen Zusammenhang. [3]

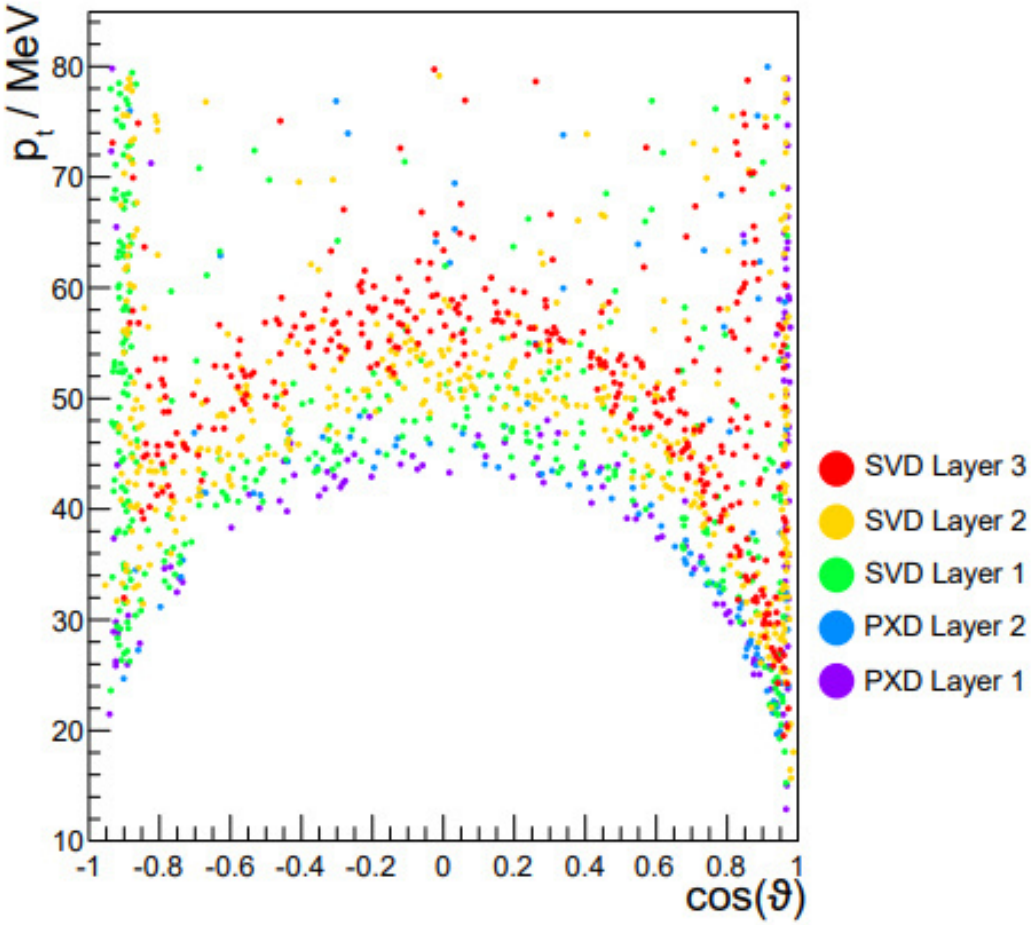


Abb. 2.4: Langsame Pionen in den innern Schichten des Detektors [3]

3 Datenanalyse

3.1 Statistische Grundlagen

Um die Daten auswerten zu können, reicht es nicht, eine einzelne statistische Methode anzuwenden. Vielmehr ist es nötig, mehrere Auswertungsmethoden anzuwenden. In diesem Kaptiel werden die wichtigsten Methoden vorgestellt, die für die Auswertung benutzt wurden.

3.1.1 Studentisierung

Die Studentisierung ist eine mathematische Transformation, bei der eine empirische Verteilung X durch Subtraktion des arithmetischen Mittels und Division der empirischen Standardabweichung zentriert und somit vergleichbar mit anderen Verteilung wird. Dies geschieht durch folgende Rechnung:

$z_i = \frac{x_i - \bar{x}}{\sigma}$, wobei $X \to Z$ und $z_i \sim Z$.

Oftmals wird für diese Art der Transformation fälschlicherweise der Begriff Standardisierung verwendet. Die Standardisierung beschreibt eine Transformation, bei der eine stochastische Verteilung zentriert wird, also um den Erwartungswert und der Standardabweichung, weshalb diese terminologisch für eine Datenauswertung nicht in Frage kommt [12].

3.1.2 Boxplot

Ein Boxplot dient der Visualisierung von statistischen Lage- und Streuparametern eines univariaten Datensatzes (das heißt, dass lediglich ein Merkmal untersucht wird). Enthält ein Datensatz mehrere Merkmale, so können diese separat beobachtet und in verschiedenen Boxplots dargestellt werden.

Die Streu- und Lageparameter, die durch einen Boxplot zusammengefasst werden, sind die folgenden :

i) Der Maximalwert

ii) Der Minimalwert
iii) Der Median (2. Quartil)
iv) Das 1. und 3. Quartil)
v) Mögliche Ausreißer

Dabei stellen das 1. und 3. Quartil die Kanten einer Box dar, und der Median wird durch eine Linie dargestellt. Zudem wird eine Verbindungslinie zum Maximal- und zum Minimalwert gezogen, sofern die Abstände dieser kleiner sind als das 1,5-fache des Interquartilabstandes (also der Abstand zwischen 1. und 3. Quartil). Andernfalls werden Punkte, die nicht in diesem Intervall liegen, als Ausreißer markiert.

Der Vorteil dieser Darstellung liegt darin, dass somit die unteren bzw. oberen 25%, 50% und 75%, sowie die „mittleren“ 50%, bereinigt um die Ausreißer, visualisiert werden. [13]

3.1.3 qq-Plot

Ein qq-Plot wird verwendet, wenn die Vermutung nahe liegt, dass der untersuchte Datensatz einer bekannten Verteilung unterliegt. Der Datensatz wird dabei nach den Werten sortiert und gegenüber den Quantilen der entsprechenden Verteilung geplotet. Liegen bspw. 200 Datenpunkte vor, so wird die vermutete Verteilung in 200 Quantile unterteilt. Weisen die aufgetragenen Punkte einen linearen Zusammenhang auf, so gilt die Annahme der vermuteten Verteilung, da sich die wahre und die vermutete Verteilung dann nur noch um eine Standardisierung bzw. Studentisierung unterscheiden.

Der Nachteil bei der Auswertung mit qq-Plots ist, dass man die zu Grunde liegende Verteilung erahnen muss und somit theoretisch unendliche viele Möglichkeiten bestehen. Einfachheitshalber beschränkt sich diese Arbeit lediglich auf die Normalverteilung. [13]

3.1.4 Kerndichteschätzer

Ein Kerndichteschätzer dient ähnlich wie ein Boxplot zur Veranschaulichung eines univariaten Datensatzes. Anders als beim Boxplot, fasst der Kerndichteschätzer den

Datensatz nicht durch Kenngrößen zusammen, sondern operiert auf dem ganzen Datensatz und stellt eine Kurve dar, die die Wahrscheinlichtkeitsdichte repräsentieren soll. Dabei kann der Kerndichteschhätzer als Erweiterung eines Histogramms zu einer stetigen Funktion aufgefasst werden.

Die Kurve wird dabei so konstruiert, dass jedem Datenpunkt ein Kern zugeordnet wird (z.B. eine Gaußkurve $k(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}t^2}$, mit $\mu = 0$ und $\sigma = 1$). Nun werden die Kerne derart überlagert, dass folgende Funktion für den Kerndichteschätzer gilt: $f_h(t) = \frac{1}{hN} \sum_{j=0}^{N} k(\frac{t-x_j}{h})$. Dabei sind $x_1, ..., x_N$ die Datenpunkte, k der Kern und der Parameter $h > 0$ die Bandbreite. Diese hat maßgeblichen Einfluss auf den Kurvenverlauf: Größere Werte glätten die Kurve, wohingegen kleinere Werte die Kurve mehr "verwackeln". Durch zu große Werte können also Strukturen in der wahren Verteilung verloren gehen bzw. bei zu kleinen Werten Strukturen entstehen, die eigentlich nicht existieren. [13]

3.1.5 Bhattacharyya-Koeffizient

Bhattacharyya-Koeffizienten können verwendet werden, um ein Maß für die Trennbarkeit von zwei Klassen zu bestimmen. Der Bhattacharyya-Koeffizient untersucht dabei ein Merkmal von zwei verschiedenen Klassen. Dabei wird die Verteilung der Daten durch Bins diskretisiert.

Ist die relative Häufigkeit des Merkmals X bzw Y gegeben durch $h(X)$ bzw. $h(Y)$ und die Bins durch $i \in [1, 2, ..., N]$, dann gilt für den Bhattacharyya Koeffizienten B(X,Y):

$$B(X,Y) = \sum_{i=1}^{N} \sqrt{h_i(X) \cdot h_i(Y)}.$$

Geometrisch lässt sich B(X,Y) auch leicht interpretieren: hierbei handelt es sich um das Skalarprodukt $< \sqrt{h(X)}, \sqrt{h(Y)} >$.
Folglich muss gelten:
$B(X,Y) = |\sqrt{h(X)}| \cdot |\sqrt{h(Y)}| \cdot cos(\alpha)$, wobei α der einschließende Winkel zwischen den beiden Vektoren ist.
Zudem gilt für $|h(X)|$, bzw. analog für $|h(Y)|$:

$|\sqrt{h(X)}| = \sqrt{\sqrt{h_1(X)}^2 + \sqrt{h_2(X)}^2 + ... + \sqrt{h_N(X)}^2} = \sqrt{\sum_{i=1}^{N} h_i(X)} = 1,$
und somit:
$B(X,Y) = cos(\alpha)$, also: $B(X,Y) \in [0,1]$
Dabei beschreibt B(X,Y)=0 den Fall, dass die Vektoren $h(X)$ und $h(Y)$ orthogonal sind, also die Daten maximal trennbar, und $B(X,Y) = 1$, dass die Vektoren Vielfache voneinander sind und die Daten somit nicht trennbar sind. Niedrige Werte sind für die Klassifizierung also wünschenswert. [14]

3.2 Klassifikatoren

Ein Klassifikator K ist eine Abbildungen $K : \mathbb{R}^n \to L$, bei der ein n-dimensionaler Merkmalsvektor $x \in \mathbb{R}$ auf eine Klasse aus L (z.B. $L = \{-1, 1\}$) abgebildet wird. Also entscheidet K, als welche Klasse ein Objekt interpretiert wird. Beispielsweise kann ein Klassifikator K konstruiert werden, welcher untersucht, ob es sich bei einer Person $x \in M$ mit der Körpergröße x_1, dem Körpergewicht x_2 und der Schuhgröße x_3 um einen Mann oder eine Frau handelt.

Um eine solche Abbildung zu finden, muss der Klassifikator "trainiert"werden, das heißt, dass auf Grund von vorhandenen Trainingsdaten ein Algorithmus angewandt wird, der ein Entscheidungskriterium für jeden Punkt im Merkmalsraum (englisch: feature Space) erstellt. Im Anschluss kann nun mit Hilfe von Testdaten geprüft werden, wie gut der Klassifikator trainiert wurde, indem mehrere Kenngrößen erhoben werden. Trainings und Testdaten werden dabei zufällig aus allen vorliegenden Daten gewählt, in dieser Arbeit ausschließlich im Verhältnis 75:25. [15]

3.2.1 Konfusionsmatrix

Um einordnen zu können, wie gut der Klassifikator funktioniert, ist es nötig verschiedene Kenngrößen, die sogenannten Metriken zu betrachten. Diese ergeben sich direkt aus der Tatsache, dass die betrachteten Objekte eine wahre Klasse und eine durch den Klassifikator vorausgesagte Klasse besitzen. Dadurch ergeben sich die vier folgenden Fälle:

1. Das Objekt besitzt die wahre Klasse A und der Klassifikator entscheidet auf Klasse A (True positive, TP)

2. Das Objekt besitzt die wahre Klasse A und der Klassifikator entscheidet auf Klasse B (False negative, FP). Diese Fehlklassifikation wird als Fehler 2. Art bezeichnet (ünter dem Radar")

3. Das Objekt besitzt die wahre Klasse B und der Klassifikator entscheidet auf Klasse A (False positive, FN) Diese Fehlklassifikation wird als Fehler 1. Art bezeichnet ("falscher Alarm")

4. Das Objekt besitzt die wahre Klasse B und der Klassifikator entscheidet auf Klasse B (True negative, TN)

Die Fehler 1. und 2. Art hängen von der Definition der Klassen A und B ab. Die Definition wird in der Regel so gewählt, dass der Fehler 1. Art schlimmere Folgen hat, als der Fehler 2. Art. Ein klassisches Beispiel hierfür ist ein gerichtliches Urteil: Klasse A wäre in diesem Fall ßchuldigünd Klasse B „unschuldig ". Tritt ein Fehler 2. Art ein, wird der Angeklagte fälschlicherweise freigesprochen, während bei einem Fehler 1. Art, eine unschuldige Person zu unrecht verurteilt wird, was gemäß den Prinzipien eines modernen Rechtsstaates wesentlich schlimmer ist. Um die Häufigkeit eines Fehlers 1. Art zu minimieren gilt deshalb der Grundsatz ïm Zweifel für den Angeklagten". [13]

Die vier verschiedenen Fälle können nun zur besseren Übersicht in einer Matrixform dargestellt werden. Daraus lassen sich nun weitere Größen ableiten, welche als Metriken bezeichnet werden:

1. Genauigkeit, $ACC = \frac{TP+TN}{TP+TN+FP+FN}$, "Korrekte Klassifizierung im Verhältnis zum gesamten Datensatz

2. Positiver Vorhersagewert, $PPV = \frac{TP}{TP+FP}$, "Korrekte Klassifizierung der Klasse A im Verhältnis zum Vorkommen der Klasse A"

3. Negativer Vorhersagewert, $NPV = \frac{TN}{TN+FN}$, "Korrekte Klassifizierung der Klasse B im Verhältnis zum Vorkommen der Klasse B"

4. Sensitivität, $TPR = \frac{TP}{TP+FN}$, "Korrekte Klassifizierung der Klasse A im Verhältnis zur Klassifizierung der Klasse A"

5. Spezifität, $TNR = \frac{TN}{TN+FP}$, "Korrekte Klassifizierung der Klasse B im Verhältnis zur Klassifizierung der Klasse B"

3.2.2 Under/- Overfitting

Die Unter- und Überanpassung, oder das, auch im Deutschen oft verwendete, Under- bzw. Overfitting, beschreibt einen Effekt, bei dem ein Datensatz zu „schlecht "bzw. zu „gut"an ein Modell angepasst wird. Explizit im Fall eines binären Klassifikators, bedeutet das, dass bei der Unteranpassung die Entscheidungsregel für die Klassifikation zu schwach trainiert wurde, sodass diese „zu grob "wirkt. Bei der Überanpassung tritt genau das Gegenteil ein: es wurde zu stark trainiert, sodass die Entscheidungsfunktion „zu präzise "wirkt. In beiden Fällen werden zu viele Testdaten falsch klassifiziert. Das Bild aus Abbilung 3.1 verdeutlicht diesen Zusammenhang. [15]

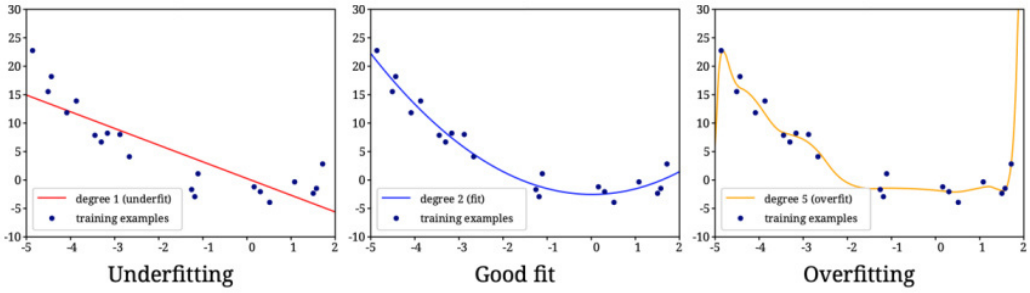


Abb. 3.1: Prinzip Über/- Unteranpassung [3]

4 Support Vector Machines

Ein häufig verwendeter Klassifikator ist das Modell der Support Vector Machines. Die Grundlegende Idee dieses Klassifikators ist, dass zwei (oder mehr) Klassen durch eine $n-1$ dimensionale Hyperebene im n dimensionalen Merkmalsraum getrennt werden. Dabei soll diese derart gewählt werden, dass der Abstand der Punkte zur Hyperebene maximal ist. Abbildung 4.1 verdeutlicht dieses Prinzip: es stehen vier Möglichkeiten zur Auswahl, zwei Klassen voneinander zu trennen. Intuitiv würde man sich für Fall d) entscheiden. [15]

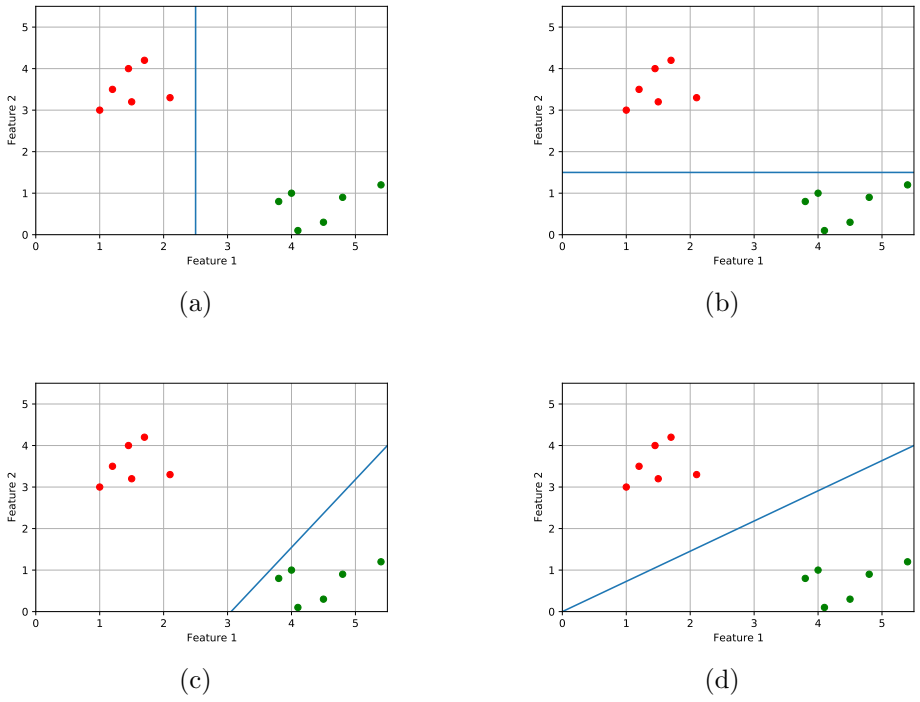


Abb. 4.1: Vier verschiedene Möglichkeiten zwei Klassen zu trennen. [16]

Nun werden zwei weitere Hyperebenen erzeugt, die parallel zur ersten Hyperebene liegen und in jeweils entgegengesetzte Richtungen zur ersten Hyperebene verschoben sind. Die Stützvektoren dieser parallelen Hyperebenen sind die namensgebenden Support Vektoren.

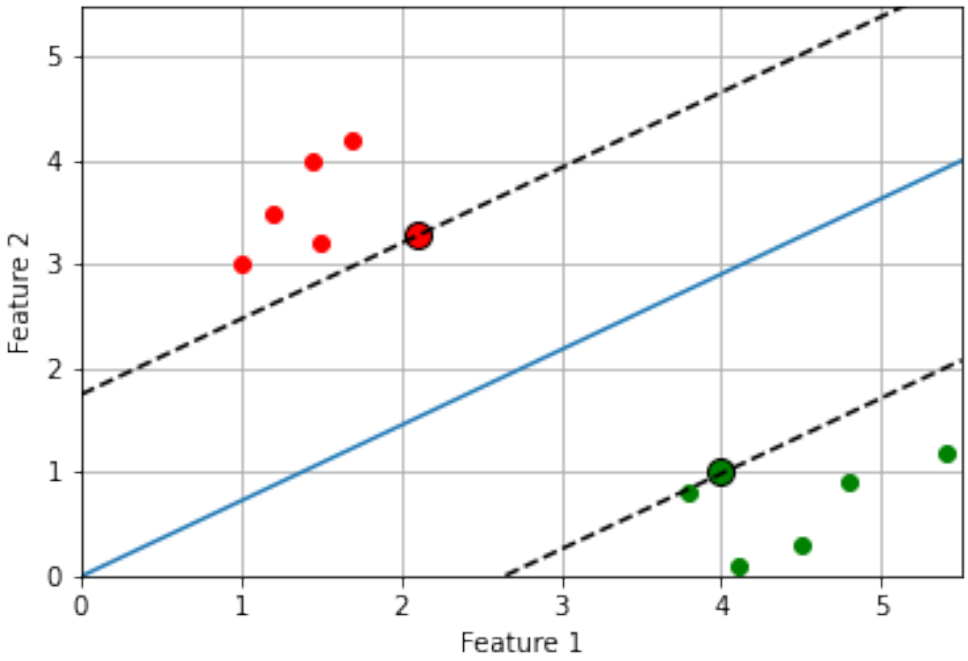


Abb. 4.2: Das selbe Klassifikationsproblem wie in Abbildung 4.1. Hier durch Support Vektoren (die markierten Punkte) und darauf stützende Hyperebenen (hier: Geraden) ergänzt [16]

4.1 Algorithmus

Es sei $M = \{(x^{(i)}, y^{(i)}) | i = 1, ..., k\} \subset \mathbb{R}^n \times L$ eine Menge aus n-dimensionalen Merkmalsvektoren $x^{(i)} \in \mathbb{R}^n$, denen zusätzlich ein Klassenlabel $y^{(i)} \in L = \{-1, 1\}$ zugeordnet wird. Bei M handelt es sich also um einen Trainings- bzw. Testdatensatz mit den Datenpunkten $i = 1, ..., k$. [15]

Nun wird eine Diskriminanzfunktion $g : \mathbb{R}^n \to L$ gesucht, mit $w \in \mathbb{R}^n$ und $b \in \mathbb{R}$, sodass: $g(w^T x^{(i)} + b) = y^{(i)}$ für möglichst viele i. Dafür wird O.B.d.A. festgelegt, dass $g(w^T x^{(i)} + b) = 1$, für $w^T x^{(i)} + b \geq 0$ und $g(w^T x^{(i)} + b) = -1$ sonst.
Geometrisch wird dies durch eine Hyperebene $\mathcal{H} = \{x \in \mathbb{R}^n | w^T x^{(i)} + b = 0\}$ realisiert, die nun die Klassen voneinander trennt. [15]

Durch die Einführung des Funktionalrandes $\gamma^{(i)} = y^{(i)} \cdot (w^T x^{(i)} + b)$ kann nun ausge-

wertet werden, wie gut eine Hyperebene gewählt wird: $(w^T x^{(i)} + b)$ beschreibt den Abstand zur Hyperebene, während $y^{(i)}$ eine korrekte Klassifizierung beschreibt. Somit gibt das Produkt an, dass ein Vektor mit großer Sicherheit korrekt (bzw. falsch) klassifiziert wird, wenn $\gamma^{(i)}$ positiv (bzw. negativ) und betraglich groß ist. [15]

Führt man nun einen Normierungsfaktor ein $||w||$, lässt sich der kleinsten mögliche Funktionalrand $\frac{\gamma}{||w||} := (\underset{i=1,...,k}{min} \gamma^{(i)}) \cdot \frac{1}{||w||}$ bestimmen, wobei der Normierungsfaktor dafür sorgt, dass die Darstellung eindeutig ist. [15]

Das Ziel des Algorithmus besteht nun darin γ zu maximieren. Dadurch wird die direkte Verbindungsstrecke der Support Vektoren minimiert, also die Klassifikation besser. Aus diesem Grund wird auch manchmal von „optimal margin classifier“ (deutsch: „Optimaler-Abstand Klassifikator“) gesprochen. Da γ durch eine Skalierung von b und w um den Faktor a auch um den Faktor a skaliert wird, setzt man nun $\gamma = 1$ und es lässt sich die Problemstellung mathematisch wie folgt formulieren:
$\underset{\gamma,w,b}{max} \frac{\gamma}{||w||} = \underset{w,b}{max} \frac{1}{||w||} \Leftrightarrow \frac{1}{2} \underset{w}{min} ||w||^2$
Dabei wurde die Minimierungsbedingung so gewählt, dass sie konvex ist. Das heißt, dass der Graph der Bedingung unterhalb einer beliebigen Verbindungsstrecke von zwei Punkten liegt. Dadurch kann ein konvexes Optimierungsproblem formuliert werden. (mehr dazu in 4.1.2). [15]

Zusammenfassend lässt sich das Problem wie folgt darstellen:

$f(w) := \frac{1}{2}||w||^2$
$g_i(w, b) := y^{(i)} \cdot (w^T x^{(i)} + b) - 1 \leq 0$
$\mathcal{L}(w, b, \alpha) := f(w) + \sum_{i=1}^{k} \alpha_i g_i(w, b)$

Nun wird ein Parametersatz (w^*, b^*, α^*) gesucht, wodurch das sogenannte primale Problem
$p^* := \underset{w,b}{min} \ \underset{\alpha}{max} \ \mathcal{L}(w, b, \alpha)$ gelöst wird. Die passenden Werte dafür zu finden kann schwer sein, weshalb das primale Problem in das sogenannte duale Problem überführt wird:
$d^* := \underset{\alpha}{max} \ \underset{w,b}{min} \ \mathcal{L}(w, b, \alpha)$.

Es gilt $p^* = d^*$, wenn die so genannten Karush-Kuhn-Tucker-Bedingungen gelten:

1. $\nabla\mathcal{L}(w^*, b^*, \alpha^*) = 0$
2. $g_i(w^*, b^*) \leq 0$
3. $\alpha_i^* \geq 0$
4. $\alpha_i^* g_i(w^*, b^*) = 0$

Bedingung 1 entspricht dabei dem Minimum von $f(w)$ unter der Nebenbedingung $g(w, b)$, mit den Lagrange-Multiplikatoren α_i.
Bedingungen 2, 3 und 4 führen so folgender Aussage: $\forall i \in \{1, ..., k\} : (g_i(w, b) = 0 \wedge \alpha_i \neq 0) \vee (g_i(w, b) \neq 0 \wedge \alpha_i = 0)$. Dabei sind die $x^{(i)}$, zu denen $\alpha_i \neq 0$ die Support Vektoren.
$\mathcal{L}(w, b, \alpha)$ lässt sich analytisch minimieren:

$$\nabla_w \mathcal{L} = w - \sum_{i=1}^{k} \alpha_i y^{(i)} x^{(i)} = 0$$

Und somit:

$$w = \sum_{i=1}^{k} \alpha_i y^{(i)} x^{(i)}$$

Analog für b:

$$\frac{\partial}{\partial b}\mathcal{L}(w, b, \alpha) = \sum_{i=1}^{k} \alpha_i y^{(i)} = 0$$

Einsetzten der beiden Ergebnisse in $\mathcal{L}(w, b, \alpha)$ liefert:

$$\mathcal{L}(w, b, \alpha) = \sum_{i=1}^{k} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{k} \alpha_i \alpha_j y^{(i)} y^{(j)} (x^{(i)})^T x^{(j)}$$

wobei dieser Ausdruck durch Variation von α nun maximiert werden muss, um d^* zu erhalten. Das kann beispielsweise über Coordinate Ascent, ein numerischer Algorithmus, realisiert werden. Dabei wird zunächst entlang einer Variablen das Maximum gesucht, im nächsten Schritt entlang der nächsten, usw. bis das Maximum iterativ erreicht wurde. [15]

Sind die α_i^* bekannt, können anschließend w^* über $w^* = \sum_{i=1}^{k} \alpha_i^* y^{(i)} x^{(i)}$ und b^* über $b^* = -\frac{1}{2} \left(\max_{i:y^{(i)}=-1} w^T x^{(i)} + \min_{i:y^{(i)}=1} w^T x^{(i)} \right)$. Die Gleichung für b^* folgt aus den Karush-Kuhn-Tucker-Bedingungen 2, 3 und 4: ist ein α_i ungleich 0, so ist $g_i(w, b) = y^{(i)}(w^T x^{(i)} + b) = 0$ und daraus folgt: $b = w^T x^{(i)}$. Da der Abstand von b zu den Support Vektoren gleich groß sein soll, wird hierfür der Mittelwert gewählt. [15]

4.2 Optimierung

Durch den in Kapitel 4.1 beschriebenen Algorithmus können insbesondere Klassifiaktionsprobleme effektiv gelöst werden, die auch „mit dem Auge“leicht gelöst werden können, also wenn zwei (bzw. mehrerere) Klassen derart separiert vorliegen, dass sich diese (kaum) „überlappen “und dadurch durch eine Gerade (lineare Hyperebene) trennbar sind. Man spricht dabei von einer linearen Klassifikation. [15]

Da allerdings oftmals Datensätze vorliegen, bei denen genau das der Fall ist, müssen sich weitere Methoden überlegt werden, um den Algorithmus so zu optimieren, dass Underfitting vermieden wird. Zwei solcher Verbesserungen werden nun vorgestellt. [15]

4.2.1 Soft Margin

Sind die Daten annähernd linear trennbar ist es sinnvoll durch die Einführung einer Kostenfunktion eine schlechte Klassifizierung zu „bestrafen “, dabei soll die „Bestrafung “proportional zum Ausmaß der Missklassifikation sein. [15]

Definiere hierfür die das Optimierungsproblem um:

$\min_{w,\xi}(\frac{1}{2}||w||^2 + C\sum_{i=1}^{k} \xi_i)$. Die Lagrangefunktion $\mathcal{L}(w, b, \xi, \alpha)$ wird analog bestimmt und für diese gilt: $\mathcal{L}(w, b, \xi, \alpha)\mathcal{L}(w, b, \alpha)$, mit dem einzigen Unterschied, dass $0 \leq \alpha_i \leq C$. Dabei muss C „per Hand “bestimmt werden und legt fest wie „hart oder weich “von b zu den Support Vektoren ist. [15]

4.2.2 Kernel Trick

Sind Daten nicht linear trennbar, ist es möglich, dass die Merkmale um ein oder mehrere Merkmale zu erweitern, ohne dabei neue Daten erheben zu müssen. Dabei verwendet man den sogenannten Kernel Trick. Dieser transformiert die Daten in einen höher dimensionalen Raum, in dem die Daten nun im Idealfall linear trennbar sind. Nach dieser Transformation kann also der Standardalgorithmus angewandt werden, um die Daten zu trennen. [15]

Betrachte dafür eine Abbildung $\phi : \mathbb{R}^n \to \mathcal{M}$, wobei das Argument von ϕ einer der n-dimensionalen Datenpunkte $x^{(i)}$ übergeben bekommt und $dim(\mathcal{M}) \geq n$.

Nachdem die Datenpunkte durch ϕ transformiert wurden, können diese nun in $\mathcal{L}(w, b, \alpha)$ eingesetzt werden: $\mathcal{L}(w, b, \alpha) = \sum_{i=1}^{k} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{k} \alpha_i \alpha_j y^{(i)} y^{(j)} \phi(x^{(i)})^T \phi(x^{(j)})$ (die Gleichungen für w^* und b^* sind entsprechend analog).

Die Transformation verursacht allerdings ein Problem: die Rechenzeit steigt mit $\mathcal{O}(n^2)$. Es lässt sich allerdings zeigen, dass:
$K(x^{(i)}, x^{(j)}) := \phi(<< x^{(i)}, x^{(j)} >, \frac{1_v}{||1_v||} >) =< \phi(x^{(i)}), \phi(x^{(j)}) >$,
sofern $K : \mathbb{R}^{n \times n} \to \mathcal{M}$ durch ein Skalarprodukt definiert und 1_v die Summe der kartesischen Einheitsvektoren (alle Komponenten sind gleich 1) ist. Dadurch reduziert sich der Rechenaufwand zu $\mathcal{O}(n)$ [17]

5 Auswertung

5.1 Datensatz

Bevor reale Daten verwendet werden können, ist es zunächst sinnvoll mit einen simulierten Datensatz zu arbeiten. Dabei wurden über eine Monte-Carlo-Simulation zwei Datensätze erzeugt: Langsame Pionen und Strahlungshintergrund (im folgenden Background).
Jeder dieser Datensätze besteht aus insgesamt 86 Merkmalen. Diese sind

1. Das Label
2. Die Total Cluster Charge
3. 9x9 Matrix
4. x-, y- und z-Position der Events

Dabei entspricht der Total Cluster Charge (kurz: TCC) der Summe aller Matrixkomponenten, welche die Ladung in dem entsprechenden Pixel beschreiben, wobei es vorkommen kann, dass einige Pixel den Wert 0 annehmen, also dass dort kein Event stattfindet. Des Weiteren sind alle dieser Datensätze mehrere GB groß und besitzen über mehrere Millionen solche 86 dimensionalen Datenpunkte. Da Support Vector Machines sehr anfällig gegenüber starkes rauschen sind, also zur Überanpassung neigen, wurden die Datensätze so gekürzt, dass mit maximal 50000 Datenpunkten gearbeitet wurde. Dabei wurde eine zufällige Menge des Datensatzes gewählt, sodass die Daten trotzdem repräsentativ bleiben. Der Performance-Unterschied in Abhängigkeit der Datengröße wird, ebenso wie die Abhängigkeit der zufällig gewählten Menge, später noch untersucht.

Da es, wie bereits erwähnt, vorkommen kann, dass einige Werte der 9x9 Matrix gleich 0 sind, ist es sinnvoll nicht alle Merkmale auszuwerten, sondern diese zu reduzieren. Dabei bietet es sich in, wie in 5.2 zu sehen ist, nur die (3,3)-Komponente, also den mittleren Wert der Matrix, zu berücksichtigen. Zusätzlich können die x- und y-Position auch mit $r = \sqrt{x^2 + y^2}$ und $\phi = arctan(\frac{y}{x})$ in Polarkoordinaten überführt

werden, da diese ebenso wie die kartesischen Koordinaten „sinnvoll “geometrisch interpretierbar sind und die Möglichkeiten der Merkmalskombinationen erhöhen.

5.2 Statistische Auswertung

Um einen groben Überblick der Verteilung des Pixelmusters zu bekommen, wurden Pixelplots erzeugt, die die normierten logarithmierten Summen dieser Matrixkomponenten darstellen. Dabei wurde zuerst normiert und dann logarithmiert, sodass der größtmögliche Wert durch eine 0 dargestellt wird und die kleinst möglichen Werte farblos bleiben, da dies $log(0) = -\infty$ entspricht. Die Pixelplots wurden dabei bei einer Datensatzgröße von 10000 erzeugt, um zu verhindern, dass die Summe für ein Pixel 0 entspricht und folglich einen endlichen Logarithmuswert zugewiesen bekommt. Abbilung 5.1 zeigt diese Pixelplots. Die Darstellungen lassen vermuten, dass die Daten sehr stark von dem mittleren Wert abhängen, weshalb statt der 9×9 Matrix der Haupt- und Nebenwert als Merkmal eingeführt werden. Diese beschreiben die Ladung im Punkt $(3, 3)$ bzw. die Summe der Ladungen aller anderen Punkte. In Kombination mit den in 5.1 diskutierten Merkmalen ergeben sich nun insgesamt acht Merkmale.

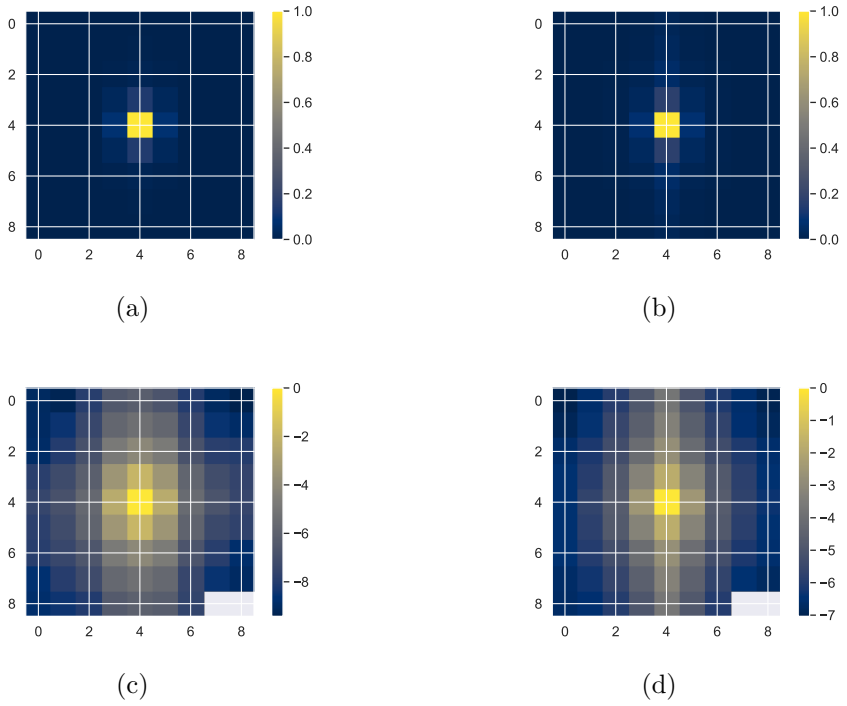


Abb. 5.1: Verteilung des Pixelmusters a) der langsamen Pionen und b) des Backgrounds, sowie die logarithmische Darstellung c) der langsamen Pionen und d) des Backgrounds [16]

Als nächstes wurden die Bhattacharyya-Koeffizienten der verwendeten Merkmale, bei einer Datensatzgröße von jeweils 2000, berechnet. Dafür wurden die Verteilungen zunächst studentisiert. Da diese, wie in 3.1.5 beschrieben, von der Anzahl der verwendeten Bins abhängen, wurden verschiedene Bingrößen gewählt und in den folgenden Tabellen zusammengefasst:

Merkmal	$B_{100}(X,Y)$	$B_{500}(X,Y)$	$B_{1000}(X,Y)$	$B_{5000}(X,Y)$
Hauptwert	0,94	0,35	0,18	0,07
Nebenwert	0,86	0,82	0,80	0,73
TCC	0,93	0,87	0,86	0,49
x	0,91	0,65	0,36	0,05
y	0,80	0,64	0,31	0,05
z	0,81	0,83	0,71	0,18
r	0.14	0,17	0,09	0,02
ϕ	0.93	0,85	0,74	0,21

Die Werte sind, je nach Anzahl der Bins, am kleinsten für den Hauptwert, x, y, z und r. Insbesondere r erreicht bei allen vier verschieden Binanzahlen einen sehr kleinen Wert. Deshalb sind Klassifizierungsversuche mit diesen fünf Merkmalen wesentlich besser geeignet, als mit den anderen drei Merkmalen (Nebenwert, TCC und ϕ), welche durchgehend hohe Bhattacharyya-Koeffizienten besitzen.

Dieses Verhalten lässt sich insbesondere auch gut erkennen, wenn die verschiedenen Verteilungen grafisch ausgegeben werden. Hierfür bieten sich Darstellung über einen Kerndichteschätzer an (Abbildung 5.2). Insbesondere für den Abstand r ist der geringe Koeffizient sehr gut anhand der Grafik erkennbar. Auch der hohe Koeffizient der Winkelverteilung ϕ ist deutlich erkennbar.

Für eine bessere grafische Darstellung wurden zudem qq-Plots erstellt. Diese dienen ergänzend zum Kerndichteschätzer für ein besseres Verständnis der Verteilung. Diese sind in Abbildung 5.3 dargestellt. Es lässt sich feststellen, dass mit Ausnahme des Hauptwertes und des Abstandes alle anderen Merkmale in der Mitte annähernd normalverteilt sind.

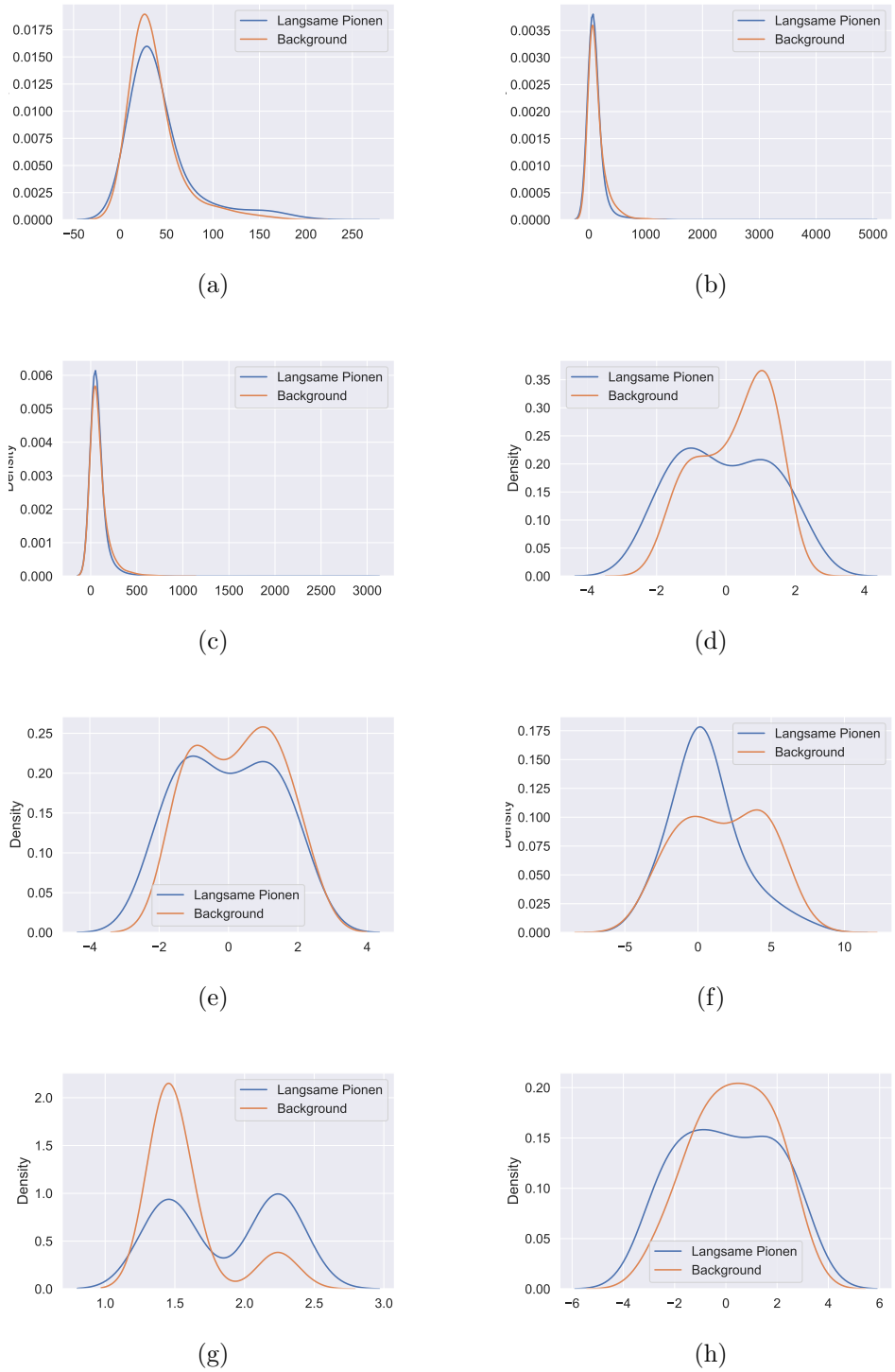


Abb. 5.2: Kerndichteschätzer a) des Hauptwertes, b) des Nebenwertes, c) der Total Cluster Charge, d) der x-Position e) der y-Position, f) der z-Position, g) des Abstandes r und h) des Winkels ϕ [16]

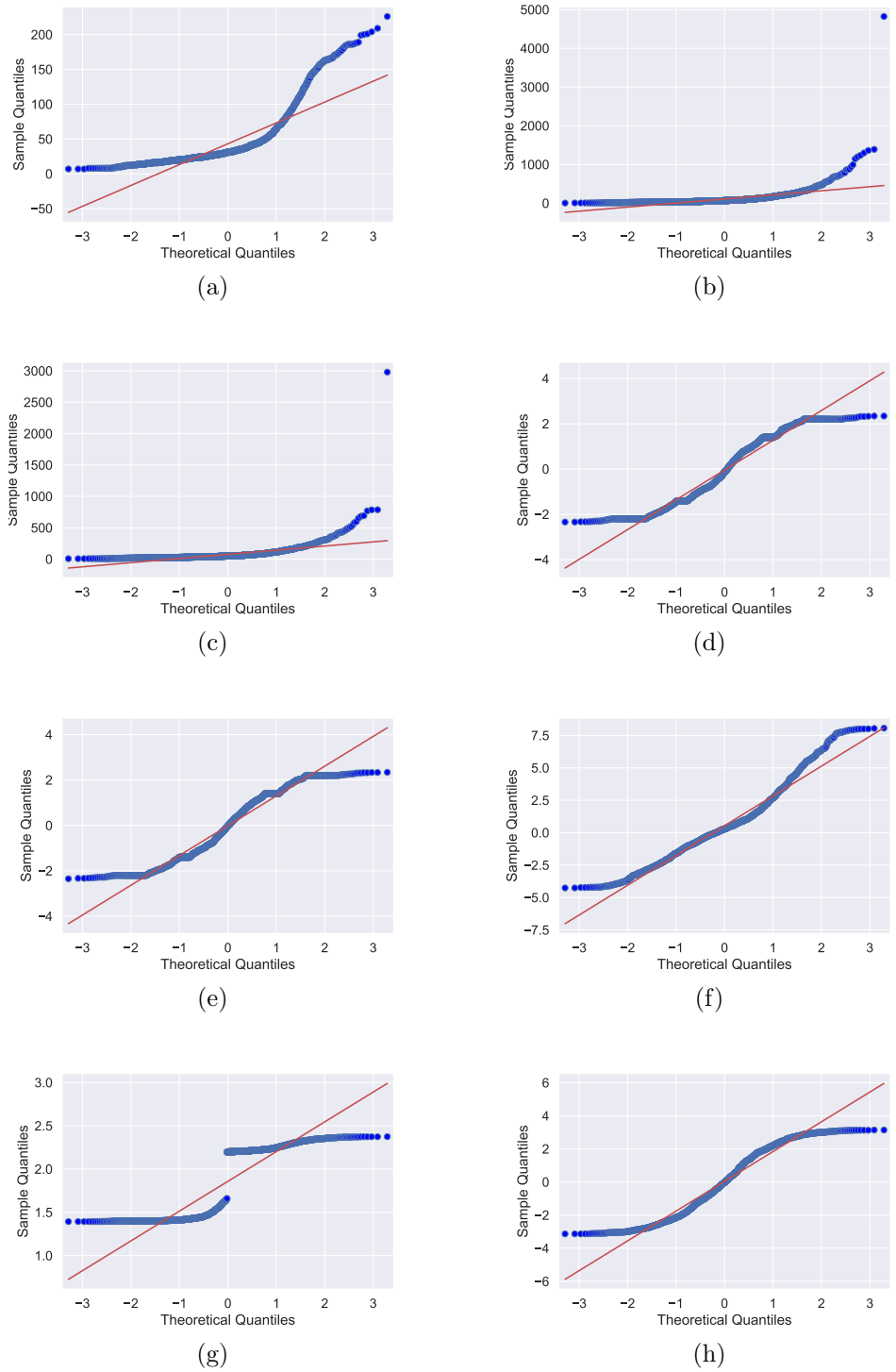


Abb. 5.3: qq-Plots a) des Hauptwertes, b) des Nebenwertes, c) der Total Cluster Charge, d) der x-Position e) der y-Position, f) der z-Position, g) des Abstandes r und h) des Winkels ϕ [16]

5.3 Klassifizierung

Da die Wahl der Parameter für die Support Vektoren bzw. für diesen Datensatz offensichtlich nicht trivial sind, ist es nötig verschiedene Parametersätze auszuprobieren, um die bestmögliche Performance zu erzielen. Dieser Parametersatz ist nicht zu verwechseln mit dem optimalen Parametersatz aus Abschnitt 4.2, sondern umfasst explizit in unserem Fall:

1. Wahl des Datensatzes
2. Die Größe des Datensatzes
3. Den Kernel
4. Die Merkmale
5. Die Parameter γ und C für den Kernel und die Softmargin

Daraus ergeben sich, die Testreihen aufzubauen sind, um die empirisch besten Support Vektoren zu finden. Für jede Testreihe wurden die Metriken aus 3.2.1 bestimmt.

Zunächst wurde für ausgewählte zweidimensionale Merkmalskombinationen eine Klassifizierung durchgeführt. Dabei waren die exakten Werte vorerst nicht von Interesse (mehr dazu in den folgenden Unterkapiteln), sondern vielmehr die geometrische Verteilung und wie der Algorithmus die bei Klassen jeweils von einander trennt. Die dazugehörigen Graphen sind in Abbildung 5.4 dargestellt.

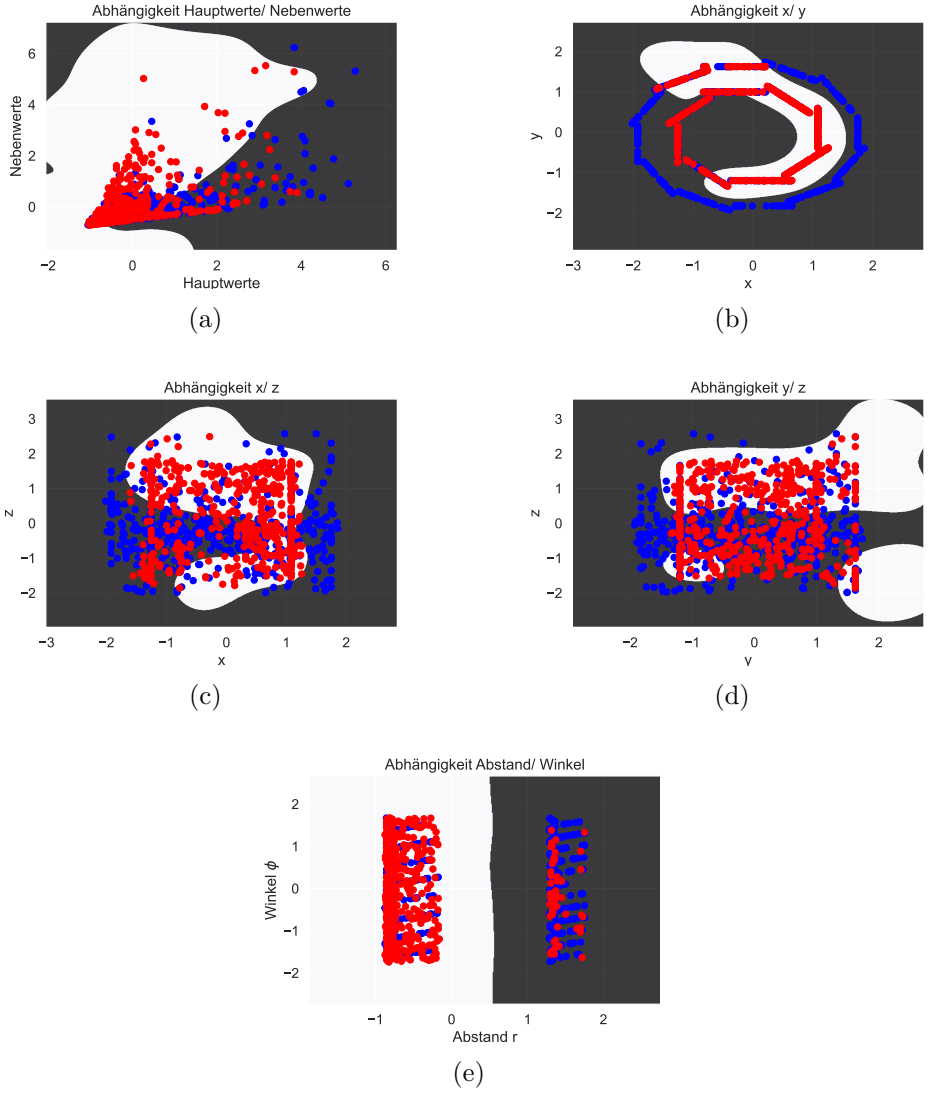


Abb. 5.4: Grafische Darstellung der Klassifizierung für ausgewählte Merkmalkombinationen. Dabei werden die langsamen Pionen durch blaue und der Background durch rote Punkte dargestellt. Punkte die in der schwarzen Fläche liegen, werden als langsame Pionen, in der weißen Fläche, werden als Background klassifiziert [16]

Es ist deutlich zu erkennen, wie großflächige Zonen entstehen, welche die langsamen Pionen vom Background trennen. Trotzdem sind viele Datenpunkte der beiden Klassen stark überlagert, wodurch auch viele Fehlklassifizierungen durchgeführt werden.

5.3.1 Wahl des Datensatzes

Um zu prüfen, ob die erzielten Ergebnisse unabhängig von der Wahl des Datensatzes sind, werden insgesamt 20 verschiedene zufällige Mengen des gesamten Datensatzes gewählt und darauf der Algorithmus angewandt. Die Ergebnisse werden als Boxplots in Abbildung 5.5 dargestellt.

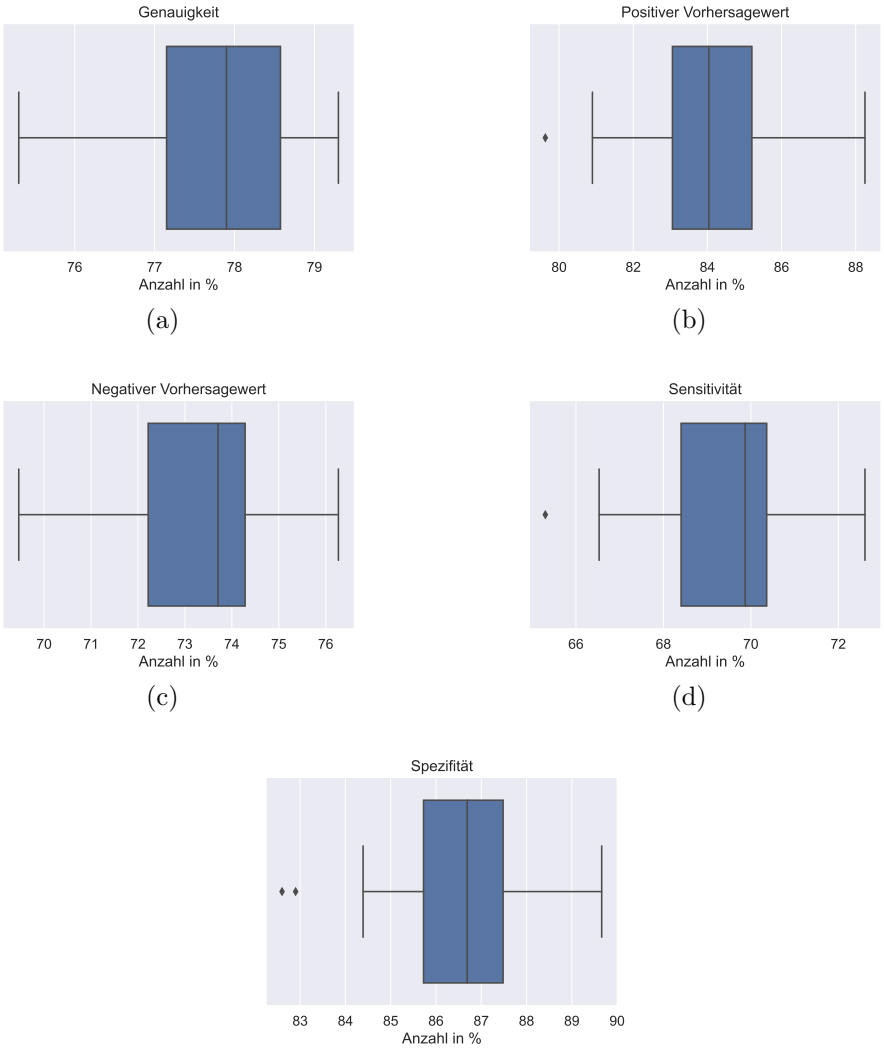


Abb. 5.5: Boxplots a) der Genauigkeit, b) des positiven Vorhersagewertes, c) des negativen Vorhersagewertes, d) der Sensitivität und e) der Spezifität [16]

Den Boxplots ist zu entnehmen, dass bei allen Metriken die Werten um etwa 6% schwanken, bzw. die mittleren 50% um etwa 3% schwanken. Die Metriken, die in anderen Testreihen bestimmt werden können, also teilweise stark von der „eigentlichen “Metrik abweichen. Um dahingehend mögliche Fehleinschätzungen vorzubeugen, wurden für die folgenden Testreihen jeweils alle Werte fünf mal berechnet und davon jeweils der Mittelwert gebildet.

5.3.2 Datensatzgröße

Wie in Abschnitt 5.1 erwähnt, ist es nötig den Datensatz zu reduzieren, um eine etwaige Überanpassung zu verhindern. Um zu testen, wie die Klassifizierung von der Datensatzgröße abhängt, werden 7 verschiedene Größen getestet: 500, 1000, 2000, 5000, 10000, 20000 und 50000. Es wurde hierfür die Merkmalskombination (Hauptwert, x, y, z, r) gewählt, da diese fünf Werte die niedrigsten Bhattacharyya-Koeffizienten aufweisen konnten. Die Ergebnisse für die Metriken sind in Abbildung 5.6 als Plot gegenüber der Datengröße als 10er-Logarithmus aufgetragen.

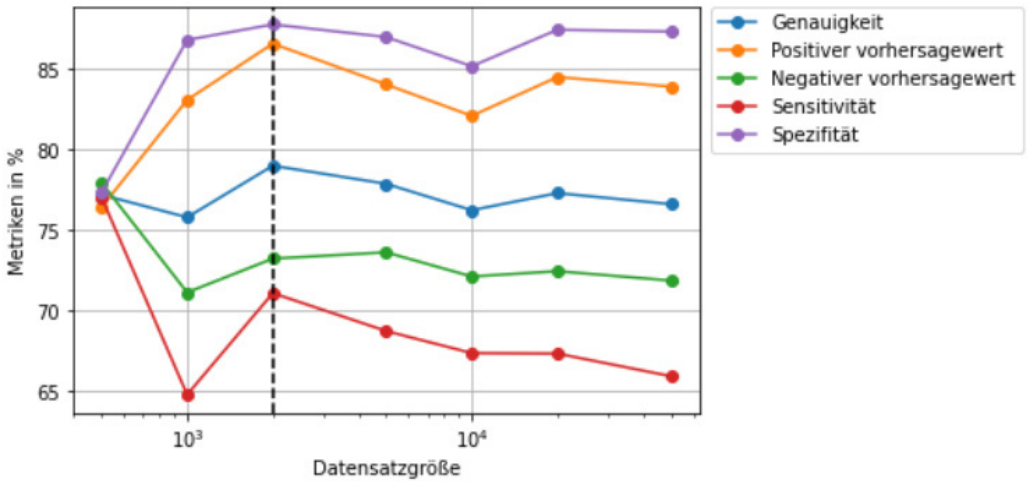


Abb. 5.6: Abhängigkeit der Metriken von der Datensatzgröße. Die schwarze gestrichelte Linie dient zur Orientierung des optimalen Wertes [16]

Wie zu erwarten war, erzielen Werte mit einem kleinen Datensatz keine hohe Genauigkeit, da hier eine Unteranpassung eintritt. Ebenso nimmt die Genauigkeit ab, wenn der Datensatz zu groß gewählt wird, da hier eine Überanpassung vorliegt.

Die besten Werte wurden für $N = 2000$ erzielt.

5.3.3 Kernel

Die Wahl des richtigen Kernels ist mitunter eines der größten Herausforderungen, wenn man mit Support Vektoren arbeitet. In dieser Arbeit wurde mit dem rbf-Kernel (radial basis function, siehe Abschnitt 3.1.4) gearbeitet, da dieser der wohl am häufigsten verwendete Kernel ist, sowie exemplarisch mit einem linearen Kernel, einem Polynom-Kernel und einem Sigmoid-Kernel (tanh-Funktion). Die Ergebnisse sind in Abbildung 5.7 als Histogramm dargestellt.

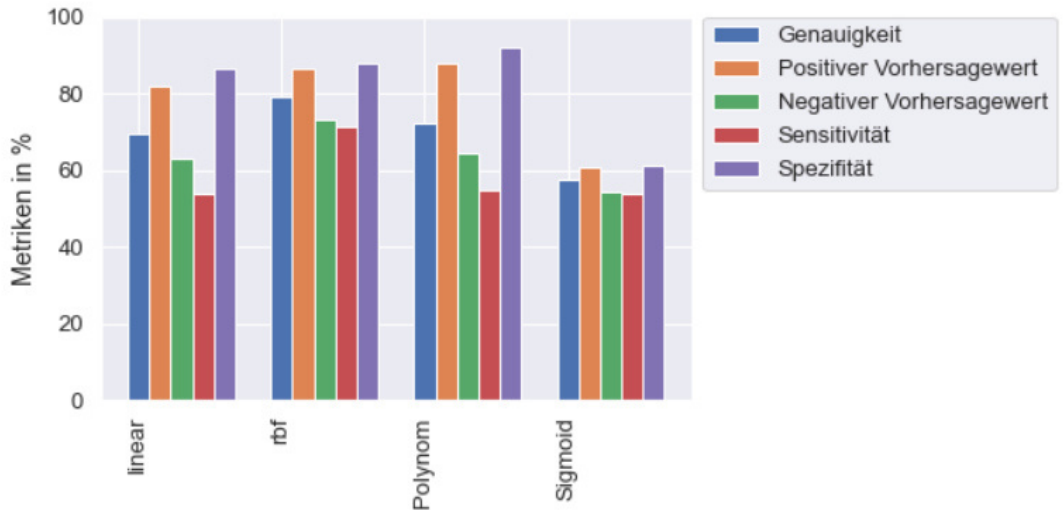


Abb. 5.7: Metriken für verschiedene Kernels [16]

Wie zu vermuten war, erzielt der rbf-Kernel eine weitaus bessere Performance, als der lineare Kernel und der Sigmoid-Kernel, da diese zur Unteranpassung neigen. Der Polynom-Kernel ist vergleichsweise besser als der lineare und der Sigmoid-Kernel, jedoch trotzdem schlechter als der rbf-Kernel, weshalb dieser in den folgenden Testreihen verwendet wird.

5.3.4 Merkmale

Es ist zu vermuten, dass die Durchführung bei zu vielen Merkmale eine Überanpassung hervorrufen wird, weshalb sich dazu entschieden wurde mehrere verschieden Merk-

malskombination auszuprobieren. Da bei 86 Merkmalen sich kombinatorisch zu viele Möglichkeiten ergeben, wurden einige wenige „sinnvolle “Kombinationen ausgewählt, welche sich auf Grund der niedrigen Bhattacharyya-Koeffizienten gewählt wurden. Die Ergebnisse sind in Abbildung 5.8 als Histogramm dargestellt.

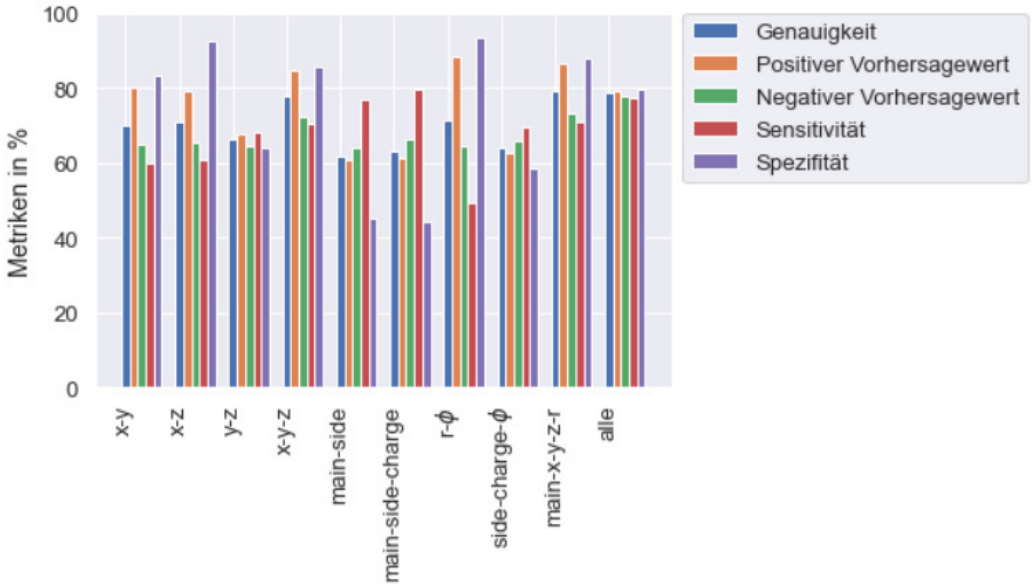


Abb. 5.8: Metriken für verschiedene ausgewählte Merkmalskombinationen [16]

Die Vermutung, dass Kombinationen mit Merkmalen, welche einen niedrigen Bhattacharyya-Koeffizienten aufweisen, besser klassifizieren bestätigt sich. Dennoch wurden auch gute Werte für alle Merkmale in Kombination erzielt. Hierbei ist die Genauigkeit etwas geringer, die anderen Metriken, liegen allerdings näher beieinander.

5.3.5 Kernel- und Softmarginparameter γ und C

Wie gut oder schlecht ein Kernel funktioniert, hängt maßgeblich von den dazugehörigen Parametern ab. Für den rbf-Kernel beschränken sich diese auf nur einen Parameter, welcher analog zur Erklärung des Kernels des Kerndichteschätzers in Abschnitt 3.1.4 auch hier mit γ bezeichnet wird. Zudem ist auch der Parameter C für die Softmargin wichtig für die Klassifizierung, da hierdurch angegeben wird, wie „hart “klassifiziert wird

Für diese Testreihe wurden die beiden Parameter γ und C variiert, während der jeweils andere Parameter einen festen Wert zugeteilt wurden. Die Ergebnisse sind in Abbildung 5.9 und Abbildung 5.10 dargestellt. Verwendet wurde hier die Kombination aus allen Merkmalen, bei einer Datensatzgröße von 2000

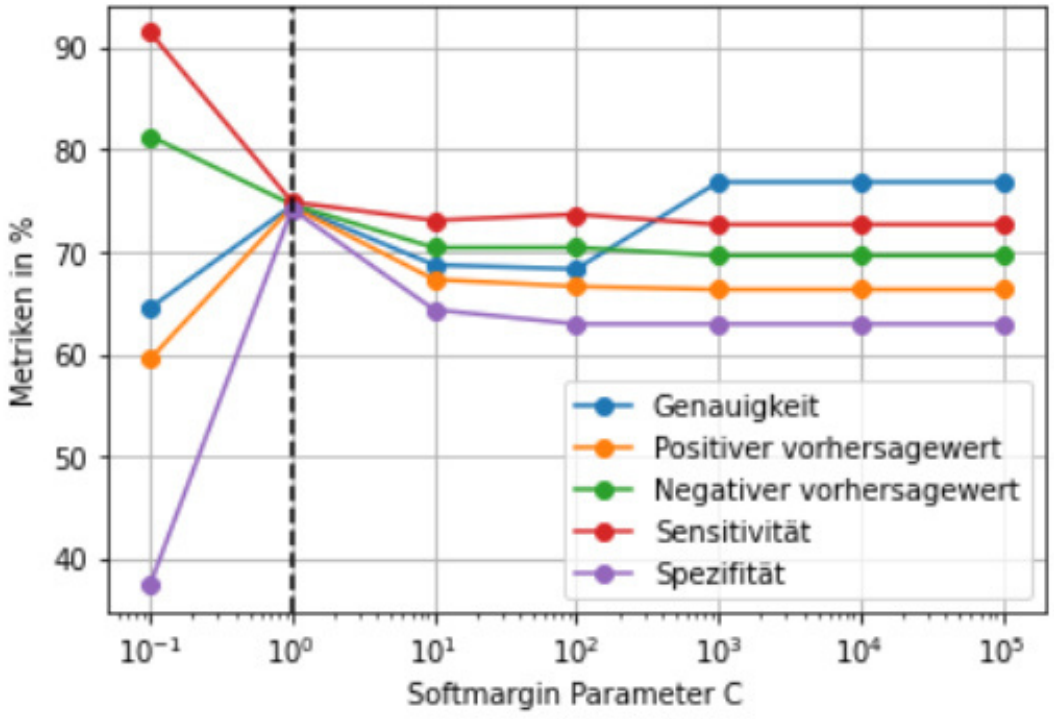


Abb. 5.9: Abhängigkeit der Metriken von dem Softmargin Parameter C. Die schwarze gestrichelte Linie dient zur Orientierung des optimalen Wertes [16]

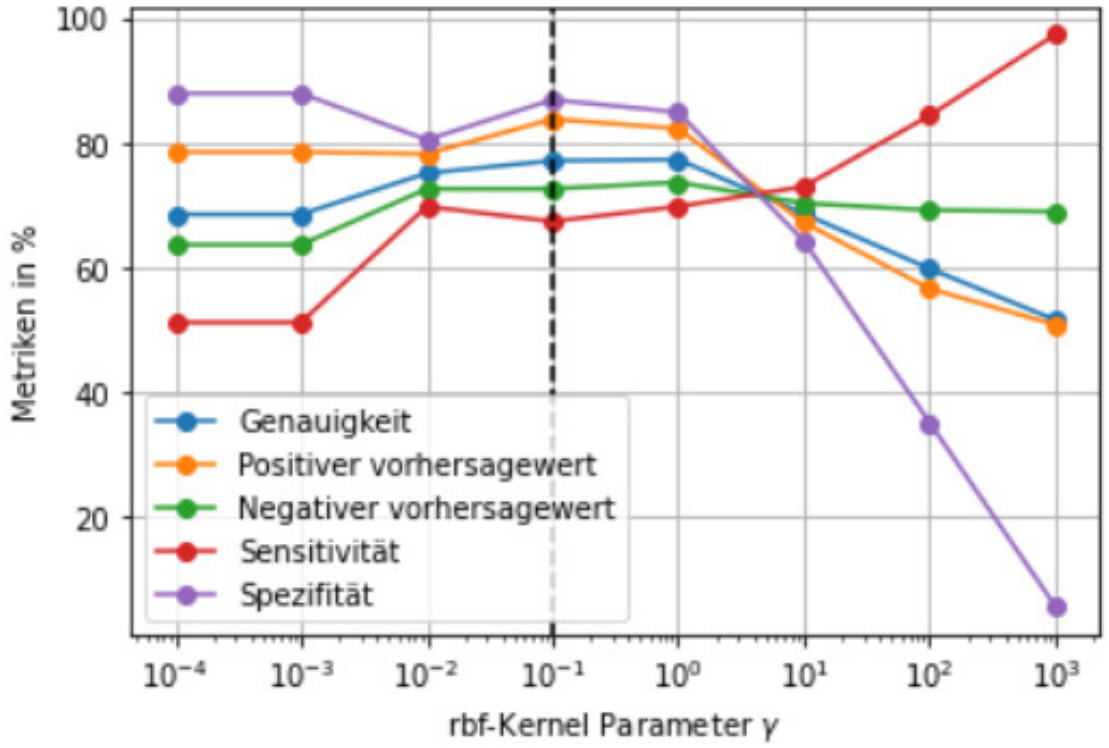


Abb. 5.10: Abhängigkeit der Metriken von dem rbf-Kernel Parameters γ. Die schwarze gestrichelte Linie dient zur Orientierung des optimalen Wertes [16]

Es ist zu sehen, dass die Metriken durch zu hohe Werte des Parameters γ, stärker streuen, und für niedrige Werte eher näher zusammenliegen, während bei dem Soft-margin Parameter C genau das Gegenteil zutrifft.

5.4 Zusammenführung der Testreihen

Berücksichtigt man nun die Ergebnisse aus allen Testreihen, können nun die optimalen Bedingungen für die Klassifizierung zusammengeführt werden. Dabei werden durch fünffache Ausführung und Bildung des jeweiligen Mittelwertes eine möglicherweise starke Abweichung, abhängig von der zufälligen Auswahl des Datensatzes, reduziert. Die gewählten Parameter entsprechen:

1. Datensatzgröße: $N = 2000$

2. Merkmalskombination: Hauptwert, x, y, z, r oder alle Merkmale
3. Kernel: rbf-Kernel, mit $\gamma = \frac{1}{10}$ und $C = 1$

Für die Werte ergeben sich dabei (Kombination 1):

1. Genauigket: $78,00\%$
2. Positiver Vorhersagewert: 82.65%
3. Negativer Vorhersagewert: 72.99%
4. Sensitivität: $68,46\%$
5. Spezifität: $85,57\%$

bzw. (Kombination 2):

1. Genauigket: $77,10\%$
2. Positiver Vorhersagewert: 80.91%
3. Negativer Vorhersagewert: 74.11%
4. Sensitivität: $71,06\%$
5. Spezifität: $83,17\%$

Johannes Bilk konnte mit den von ihn verwendeten neuronalen Netzen eine Genauigkeit von, je nach Modell, bis zu 83% erreichen. Diese Werte konnten durch die in dieser Arbeit verwendeten Support Vector Machines nahezu erreicht werden.

6 Ausblick

Es wurden einige statistische Methoden angewandt, um die simulierten Daten zu untersuchen. Dabei konnten einige Interessante Eigenschaften bezüglich der Verteilungen der einzelnen Merkmalen feststellen. Diese Ergebnisse erwiesen sich für die Auswertung der Supportvektoren als nützlich und können auch bei anderen für die Optimierung der Metriken bei anderen Algorithmen hilfreich sein.

Johannes Bilk konnte mit den von ihn verwendeten neuronalen Netzen eine Genauigkeit von, je nach Modell, bis zu 83% erreichen. Diese Werte konnten durch die in dieser Arbeit verwendeten Support Vector Machines nahezu erreicht werden. Des Weiteren konnten bei der Klassifizierung geometrische Entscheidungsgrenzen grafisch dargestellt werden, welche von Vorteil beim Verständnis sowie bei der Planung zukünftiger Machine Learning bzw. Deep Learning Algorithmen sein können.

Es ist nicht auszuschließen, dass durch Ändern der Parameter dieses Experiments, wie beispielsweise die Wahl eines geeigneteren Kernels, bessere Resultate erzielt werden können. Mit einer Genauigkeit von bis zu 78% sind die Ergebnisse dieser Arbeit noch nicht gut genug, um in Betracht gezogen zu werden, für eine tatsächliche Implementierung.

7 Quellenverzeichnis

1. Ying Li, Cai-Dian Lu, Recent Anomalies in B Physics, arXiv, 2018
2. Bevan, et al., The physics of the B factories, Springer Nature, 2017
3. Johannes Bilk, Jens Sören Lange, Employing Deep Learning to Find Slow Pions in the Pixel Detector in the Belle II Experiment, Masterthesis, Gießen, 2021
4. Wikipedia: Belle-II-Experiment, https://de.wikipedia.org/wiki/Belle-II-Experiment, letzter Aufruf: 23.02.2022
5. Stephanie Käs, Jens Sören Lange, Multiparameter Analysis of the Belle II Pixeldetector's Data, Bachelorthesis, Gießen, 2019
6. Bogdan Povh, et al. Teilchen und Kerne: Eine Einführung in die physikalischen Konzepte, Springer Verlag, 2014
7. Jens Sören Lange, Vorlesung: Höhere Teilchenphysik, Sommersemester 2021, JLU Gießen
8. Wolfgang Demtröder, et al., Experimentalphysik 4. Heidelberg : Springer Verlag, 2010
9. Claudia Höhne, Vorlesung: Höhere Hadronen- Schwerionen- und Kernphysik, Wintersemester 2021/22, JLU Gießen
10. KEK. Electrons and Positrons Collide for the first time in the SuperKEKB Accelerator: https://www.kek.jp/en/newsroom/2018/04/26/0700/, letzter Aufruf 12.01.2022
11. Wikipedia: Pionen, https://de.wikipedia.org/wiki/Pion, letzter Aufruf: 13.01.2022
12. Gerrit Eichner, Vorlesung: Ausgewählte statistische Verfahren mit R (R4), Wintersemester 2021/22, JLU Gießen
13. Gerrit Eichner, Vorlesung: Grundlagen der Statistik, JLU Gießen
14. Konstantinos G. Derpanis, The Bhattacharyya Measure, York University
15. Franz Cemic, Vorlesung. Machine Learning, Sommersemester 2021, THM Gießen
16. Grafiken wurden selbst erstellt
17. SVM Vs Neural Network: https://www.baeldung.com/cs/svm-vs-neural-network letzter Aufruf: 28.12.2021

Dankessagungen

Mit diesen abschließenden Worten möchte ich mich bei all den Menschen bedanken, die mich während meiner Zeit an der Uni und teilweise darüber begleitet, ermutigt und unterstützt haben.

Diese Arbeit und auch meine Freude an den Themen Datenauswertung und künstliche Intelligenz, wären niemals entstanden, wenn ich nicht durch die wöchentlichen Meetings in der Neuro Group durch die Expertise von Sören, Katharina, Johannes und Steffi hätte profitieren können. Die ganzen Gespräche und das Feedback haben mir nicht nur für diese Thesis geholfen, sondern auch dazu angeregt meinen Horizont dieser Themenfelder zu erweitern. Auch für die Geduld und das Verständnis, das mir Sören während dieser Arbeit entgegengebracht hat, möchte ich mich ganz herzlich bedanken. Mein besonderer Dank gilt auch Claudia Höhne, die sich für die Zweitkorrektur bereiterklärt hat.

Natürlich gilt mein Dank auch meiner Familie und meinen Freunden, auf die ich mich immer verlassen konnte. Und auch wenn ich hier nicht alle namentlich erwähnen kann, bin ich froh, es euch gibt und dass ihr mich auf diesen langen Weg begleitet habt.
Zu guter Letzt möchte ich meiner langjährigen Freundin Steffi danken. Frodo hätte niemals den Ring zerstören können, wenn Sam ihn nicht ununterbrochen auf den richtigen Weg geführt und den Rücken gestärkt hätte. Und ebenso wie Frodo gescheitert wäre, hätte auch ich niemals dieses Studium zu Ende bringen können, wenn ich dich nicht an meiner Seite gehabt hätte. Du warst all die Jahre über mein Sam und dafür möchte ich dir von ganzen Herzen Danken.

Selbstständigkeitserklärung

Hiermit versichere ich, die vorgelegte Thesis selbstständig und ohne unerlaubte fremde Hilfe und nur mit den Hilfen angefertigt zu haben, die ich in der Thesis angegeben habe. Alle Textstellen, die wörtlich oder sinngemäß aus veröffentlichten Schriften entnommen sind, und alle Angaben die auf mündlichen Auskünften beruhen, sind als solche kenntlich gemacht. Bei den von mir durchgeführten und in der Thesis erwähnten Untersuchungen habe ich die Grundsätze gute wissenschaftlicher Praxis, wie sie in der ‚Satzung der Justus-Liebig-Universität zur Sicherung guter wissenschaftlicher Praxis' niedergelegt sind, eingehalten. Gemäß § 25 Abs. 6 der Allgemeinen Bestimmungen für modularisierte Studiengänge dulde ich eine Überprüfung der Thesis mittels Anti-Plagiatssoftware.

Datum	Unterschrift