Friedrich-Alexander-Universität Erlangen-Nürnberg
Bachelorarbeit aus der Physik

Erweiterung der Ereignisklassifizierung des ANTARES Neutrinoteleskops mittels Random Decision Forests

Betreuerin: Frau Professor Gisela Anton

Vorgelegt von:
Thomas Kittler
09.10.2012

Erlangen Centre for Astroparticle Physics

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Erlangen, den 09.10.2011

Abstract

In dieser Arbeit wurde untersucht, ob Erweiterungen eines Algorithmus, welcher unter Verwendung von Random Desicion Forests Detektorereignisse klassifiziert, zu besseren Ergebnissen führen. Bei diesen Erweiterungen handelt es sich um das Hinzufügen von zusätzlichen Kennzahlen zu einer bereits bestehenden Menge. Die zusätzlichen Kennzahlen stammen aus drei für den ANTARES Detektor geschriebenen Modulen, jeweils eines für die Energie-, die Schauer- und die Spurrekonstruktion. Die Ergebnisse haben gezeigt, dass sich die Klassifikationssicherheit der Random Desicison Forests unter keinen der getesteten Umständen verschlechtert, die Effekte der einzelnen Module jedoch signifikant voneinander abweichen. Somit wurden bei Verwendung der Kennzahlen aus Energie- und Spurrekonstruktion deutlich höhere Sicherheiten gemessen, während die Kennzahlen aus der Schauerrekonstruktion keine signifikante Veränderung hervorrufen.

Zusätzlich wurde untersucht, welche Klassifikationssicherheiten sich mit einer wesentlich kleineren Menge an optimierten Kennzahlen erzielen lässt. Hierbei konnte festgestellt werden, dass sich mit nur sechs Kennzahlen bereits Sicherheiten erzielen lassen, die um weniger als 10% von den mit allen Kennzahlen erzielten Sicherheiten abweichen.

Inhaltsverzeichnis

Kapitel 1

Einleitung

Die Astronomie gilt als eine der ältesten Naturwissenschaften der Menschheit. Der Sternenhimmel hat die Menschen die gesamte Kulturgeschichte hindurch in nahezu allen Kulturen und Religionen begeistert - bis heute. Die Methoden der Astronomie haben sich von einfachen Beobachtungen von Sternenbahnen mit bloßem Auge bis hin zu moderner Forschung mit tonnenschweren Teleskopen in der Umlaufbahn der Erde entwickelt. Doch je weiter man in das Universum blicken kann, desto mehr unverstandene Phänomene werden entdeckt, was zu der Entwicklung von immer ausgefeilteren Techniken und Methoden führt.

Während man sich in der Vergangenheit und auch heute noch zu großen Teilen auf die Erfassung von elektromagnetischen Wellen konzentriert hat, wird seit einigen Jahrzehnten ein weiterer Ansatz verfolgt. Hierbei handelt es sich um die Neutrinoastronomie. Das Prinzip der Neutrinoastronomie ist es, anstelle von Photonen, kosmische Neutrinos zu detektieren und mit Hilfe der gewonnenen Informationen Rückschlüsse auf mögliche Quellen zu ziehen.

Eines der Experimente auf diesem Gebiet ist ANTARES (Astronomy with a Neutrino Telescope and Abyss RESearch). Das ANTARES Neutrinoteleskop befindet sich auf dem Grund des Mittelmeeres und ist auf die Beobachtung der südlichen Hemisphäre ausgerichtet. Da Neutrinos aufgrund der geringen Interarktion mit Materie nicht direkt beobachtet werden können, werden Sekundärteilchen detektiert. Mittels dieser Daten wird dann versucht mit Hilfe von aufwändigen Algorithmen Informationen über die Neutrinos zu erhalten.

Eine der größten Hürden bei ANTARES ist es aus der Menge an detektierten Daten die Ereignisse herauszufiltern, die durch Neutrinos verursacht wurden. Zur Lösung dieses Problems wird ein Ansatz verfolgt, welcher sich der Mustererkennung, einem Teilgebiet der Informatik, bedient (siehe [10]). Das Thema dieser Arbeit ist es den in [10] entwickelten Algorithmus zu erweitern und zu Untersuchen, ob dies zu besseren Resultaten führt.

Hierzu werden zunächst kurz die physikalischen Hintergründe von ANTARES

erläutert und der Aufbau des Neutrinoteleskops wird beschrieben. Anschließend werden die Prinzipien von Klassifikationen dargestellt und der theoretische Hintergrund von dem hier verwendeten Klassifikator, den „Random Decision Forests“, wird erklärt. In Kapitel 4 wird ausführlich auf die Implementierung und die verwendeten Programme eingegangen. Zusätzlich werden die bei der Evaluation der Ergebnisse verwendeten Methoden erläutert. Kapitel 5 stellt die berechneten Ergebnisse detailliert dar, während Kapitel 6 diese diskutiert. Ein Ausblick auf mögliche zukünftige Themen schließt die Arbeit ab.

Kapitel 2

ANTARES

In diesem Kapitel wird der Aufbau, die Funktionsweise und die Datenerfassung des ANTARES Neutrinoteleskops beschrieben. Des weiteren wird auf die Ursprünge hochenergetischer Neutrinos eingegangen.

2.1 Funktionsprinzip

Wie bereits erwähnt ist das Ziel von ANTARES die Detektion von hochenergetischen Neutrinos und die Identifizierung der Ursprünge dieser Teilchen. Neutrinos sind elektrisch ungeladene Teilchen, welche nur über schwache Wechselwirkung und Gravitation mit Materie wechselwirken. Die gravitative Wechselwirkung kann aber aufgrund der geringen Masse der Neutrinos ($m_\nu \leq 2\,eV$ [3]) in den meisten Fällen vernachlässigt werden. Diese Eigenschaften der Neutrinos machen den direkten Nachweis unmöglich. Man muss also Sekundärteilchen, welche durch Reaktion der Neutrinos mit Materie entstehen, nachweisen. Bei diesen Sekundärteilchen handelt es sich in den meisten Fällen um geladenen Leptonen. Das Feynmandiagramm einer Reaktion eines Myonneutrinos mit einem Neutron ist in Abbildung 2.1 dargestellt. Es handelt sich hierbei um eine „Charged Current“ (CC) Reaktion, da das Austauschteilchen ein geladenes W-Boson ist. Neutrinos können auch per „Neutral Current“ (NC) Reaktionen mit Materie wechselwirken. Bei dieser Form der Wechselwirkung ist das Austauschteilchen ein neutrales Z-Boson. Sowohl W- als auch Z-Bosonen sind Austauschteilchen der schwachen Wechselwirkung.

Aus der relativistischen Energie-Impuls-Beziehung kann man die Geschwindigkeit des Sekundärteilchens bei gegebener Gesamtenergie berechnen:

$$E^2 - c^2 p^2 = E_0^2 \tag{2.1}$$

$$\rightarrow v = \frac{\sqrt{E^2 - E_0^2} \cdot c}{E} \tag{2.2}$$

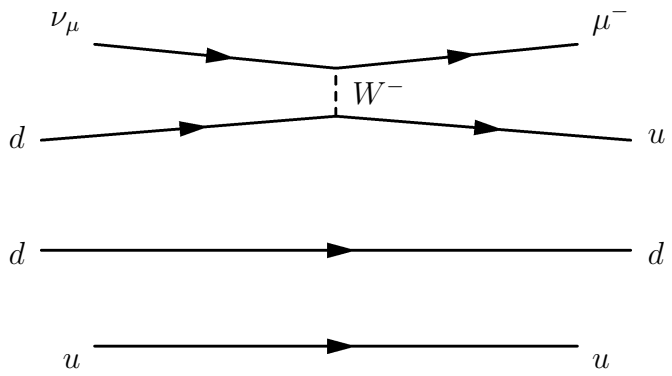

Abbildung 2.1: Reaktion eines ν_μ mit einem Neutron

Sobald die Geschwindigkeit eines geladenen Teilchens größer ist als die Lichtgeschwindigkeit in einem dielektrischen Medium, tritt der Tscherenkoveffekt auf. Hierbei handelt es sich um das Abstrahlen eines Lichtkegels entlang der Flugbahn des Teilchens. Die Energie, die ein Myon mindestens benötigt um Tscherenkovstrahlung in Wasser zu erzeugen lässt sich folgendermaßen berechnen:

$$c_{H_2O} = \frac{c_0}{n_{H_2O}} \tag{2.3}$$

$$v_{min} = \frac{c_0}{n_{H_2O}} = \frac{\sqrt{E_{min}^2 - E_0^2} \cdot c_0}{E_{min}} \tag{2.4}$$

$$\Rightarrow \frac{E_{min}^2}{E_0^2} = \frac{n_{H_2O}^2}{n_{H_2O}^2 - 1} \tag{2.5}$$

$$E_{min,\mu} \approx 1,51 \cdot 105,6\, MeV \approx 160\, MeV \tag{2.6}$$

Hierbei ist c_{H_2O} die Lichtgeschwindigkeit in Wasser, c_0 die Lichtgeschwindigkeit im Vakuum und n_{H_2O} der Brechungsindex von Wasser ($\approx 1,33$). Sobald ein Myon also eine Gesamtenergie größer als $160\, MeV$ besitzt, erzeugt es Tscherenkovstrahlung. Grund für den Lichtkegel ist die kurzzeitige Polarisation der Atome im Medium. Bei der Polarisation der Atome werden die in der Atomhülle vorhandenen Elektronen beschleunigt, worauf hin sie elektromagnetische Wellen aussenden. Normalerweise interferieren diese Wellen destruktiv mit denen ihrer Nachbarn. Sobald sich das polarisierende Teilchen mit einer Geschwindigkeit größer als die Lichtgeschwindigkeit in diesem Medium bewegt, überlagern sich die Wellen nicht mehr destruktiv. Der Winkel θ, den die Wellenfront mit der Flugbahn des Teilchens einschließt, ist abhängig von der Geschwindigkeit und somit von der Energie des Teilchens [3].

$$\beta = \frac{v}{c_0} \tag{2.7}$$

$$cos(\theta) = \frac{c_0/n}{\beta c_0} = \frac{1}{\beta n} \tag{2.8}$$

Hierbei ist v die Geschwindigkeit des Teilchens. Die Energien, welche für ANTARES interessant sind, befinden sich oberhalb von $10\,GeV$, womit $\beta \approx 1$ ist. Dies hat zur Folge, dass die Tscherenkovstrahlung unter einem konstanten Winkel von $\theta = 42°$ emittiert wird, was die Rekonstruktion von Teilchenspuren wesentlich vereinfacht, da die Energieabhängigkeit des Winkels vernachlässigt werden kann [1]. Das Funktionsprinzip des ANTARES Neutrinodetektors ist es nun diese Tscherenkovstrahlung zu detektieren und mit Hilfe der gewonnenen Informationen über Intensität, Richtung und Detektionszeitpunkt, zunächst die Flugbahn des erzeugenden Teilchens zu rekonstruieren und damit die Herkunftsrichtung des Neutrinos zu ermitteln (vgl. Abbildung 2.2).

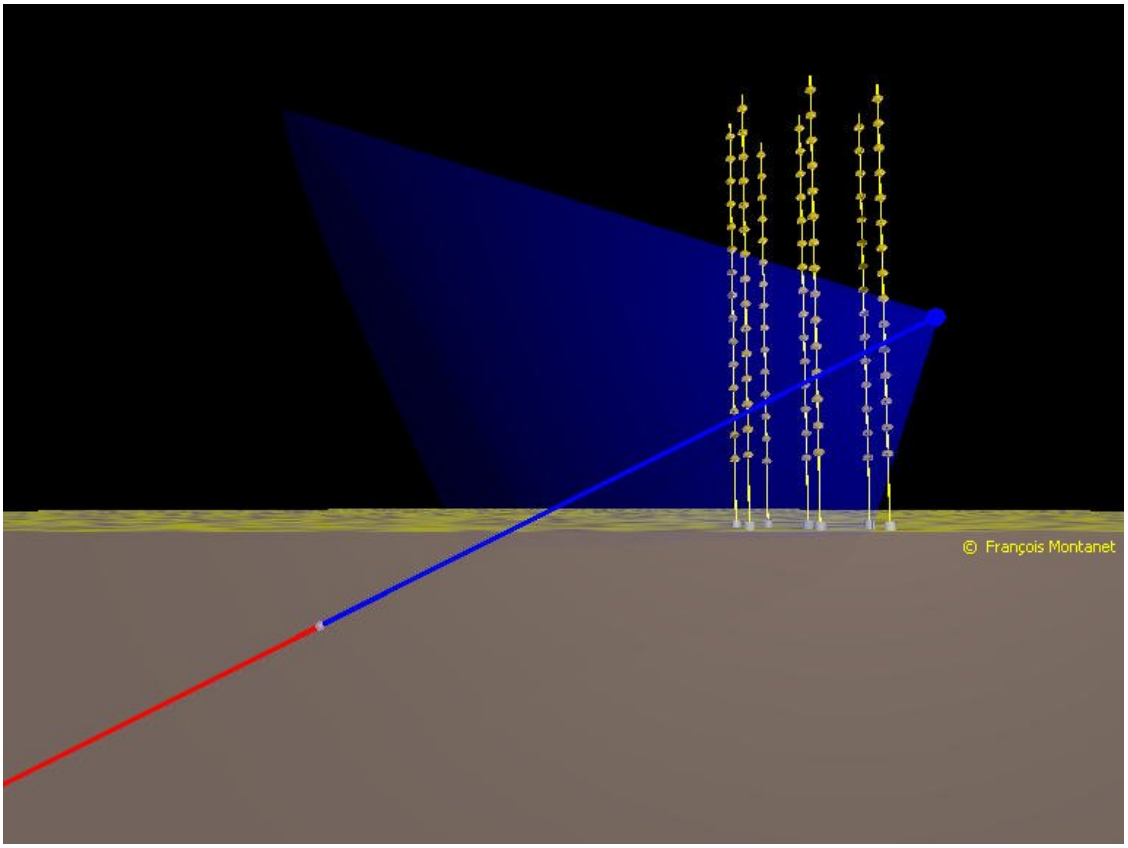

Abbildung 2.2: Von unten kommendes Neutrino (rot) reagiert innerhalb der Erdkruste (weißer Punkt) und erzeugt ein Myon (blau), welches detektiert werden kann [1]

2.2 Aufbau des Neutrinoteleskops

Das ANTARES Neutrinoteleskop besteht aus einer Anordnung von 885 sog. optischen Modulen (OM), welche sich in 2500 m Tiefe am Grund des Mittelmeeres 40 km vor der französischen Küste befindet [13]. Ein OM ist eine druckresistente Glaskugel in der ein Photomultiplier eingebaut ist, dessen Aufgabe es ist, Photonen über ein Zeitfenster von $t_{int} = 23\,ns$ zu detektieren und einen der Anzahl der detektierten Photonen entsprechend hohen elektrischen Puls zu erzeugen. Je drei OMs sind zu einem Storey zusammengefasst, wobei sich in der horizontalen je ein 120° Winkel zwischen zwei OMs befindet (Abb. 2.3). Außerdem sind die OMs um 45° nach unten geneigt, da man vor allem Teilchen messen möchte, die sich nach oben bewegen.

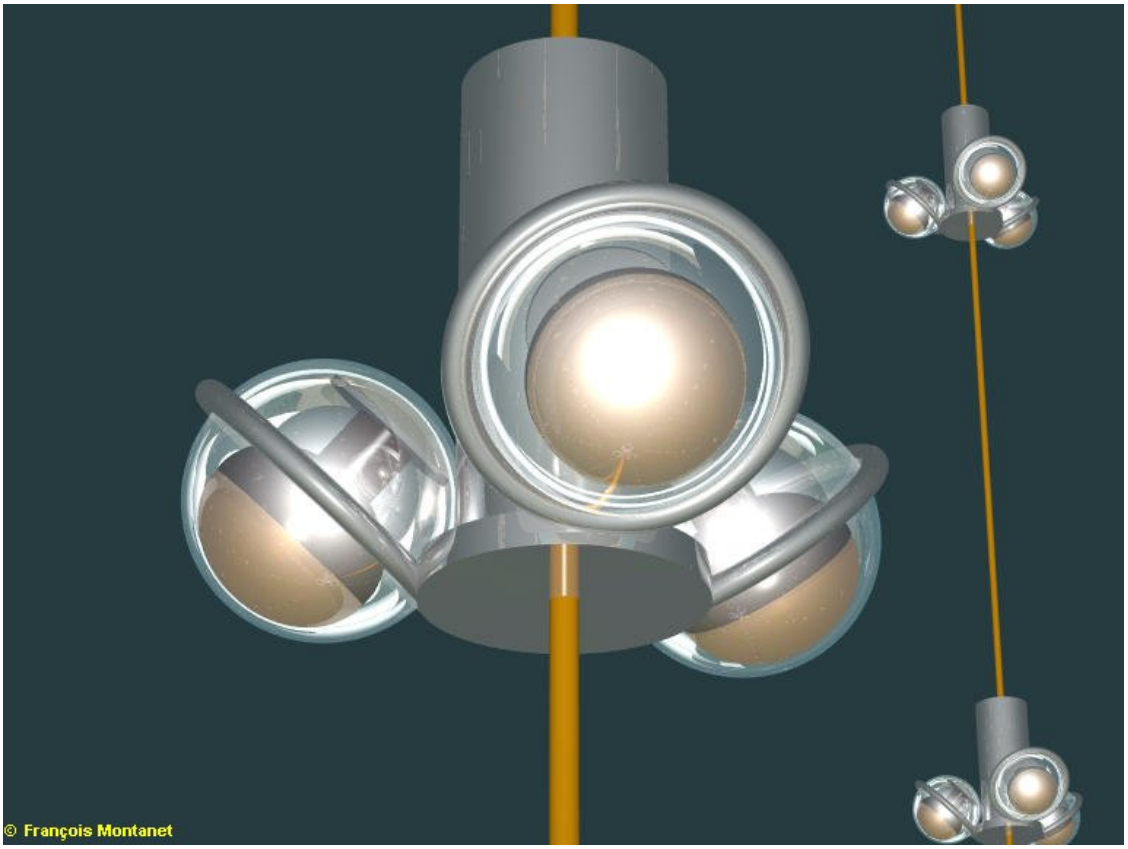

Abbildung 2.3: Storey aus drei OMs [1]

Jeweils 25 Storeys sind übereinander in einem Abstand von 14, 5 m an einer Line befestigt, wobei das erste Storey 100 m über dem Grund platziert ist. Der Detektor erreicht damit eine Höhe von ca. 450 m. Die insgesamt 12 Lines mit je 75

OMs, welche in einem Abstand von ca. $70\,m$ platziert sind (Abb. 2.4), erschließen somit ein Detektorvolumen von ca. $0,01\,km^3$ [14] und eine Oberfläche von etwa $0,1\,km^2$ [1].

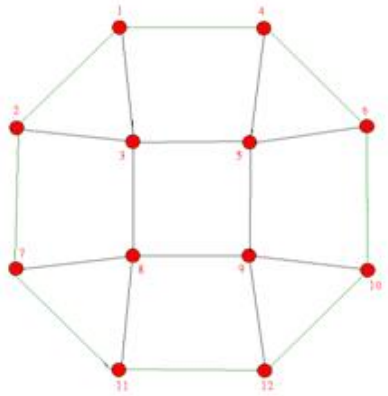

Abbildung 2.4: Anordnung der Lines von oben betrachtet [1]

Um die Position der Lines möglichst konstant zu halten sind Bojen an den oberen Enden jeder Line befestigt (Abb. 2.5). Aufgrund der Strömung neigen sich die Lines allerdings trotzdem, was eine kontinuierliche Positionsbestimmung notwendig macht. Dies wird unter anderem mittels akustischer Signale bewerkstelligt. Hierzu erzeugt ein Sender, welcher sich an einem festen Platz am Meeresgrund befindet, zu bestimmten Zeitpunkten akustische Impulse. Diese Pulse werden mittels mehrerer Hydrophone an den Lines registriert. Aus der Laufzeit der Signale kann der Abstand vom Mikrophon zum Sender bestimmt werden. Zusätzlich wurden an jeder Line Neigungsmesser und Kompasse angebracht. Aus den mit diesen Instrumenten gewonnen Informationen lässt sich die Position jeder Line bestimmen [14].

Am Grund einer jeden Line befindet sich ein Elektronik Container (SCM), in welchem unter anderem die Signale der OMs und der Mikrophone zusammenlaufen. Diese SCMs sind mittels optoelektronischer Kabel mit einer gemeinsamen Anschlussbox verbunden. Diese Anschlussbox ist dann über ein ca. $40\,km$ langes Kabel mit der Station an Land verbunden. Hier werden zunächst alle Daten zwischengespeichert bevor weitere Selektionsmechanismen in Kraft treten.

2.3 Datenerfassung und Selektion

Einer der größten Nachteile der Positionierung des ANTARES Neutrinoteleskops auf dem Grund des Mittelmeeres ist die natürliche Umgebung, welche für den größten Teil des Hintergrundrauschens verantwortlich ist. Hierbei sind vor allem

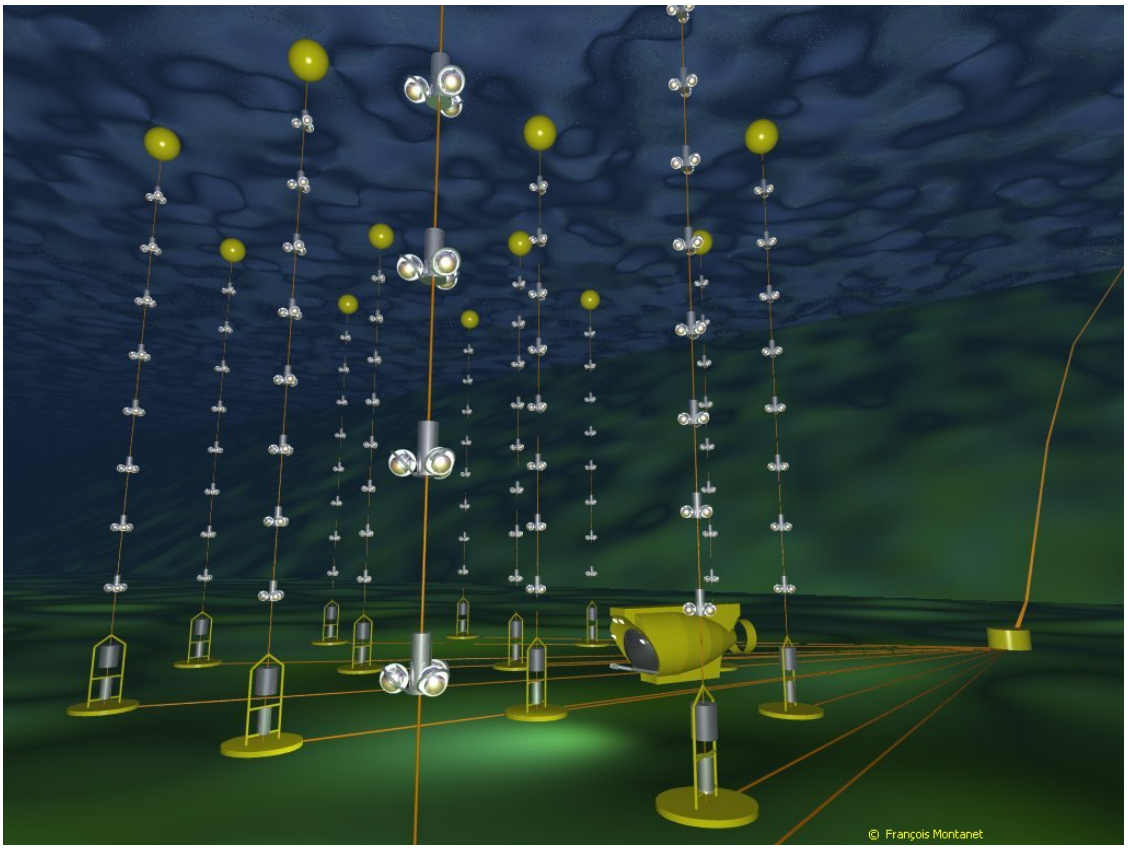

Abbildung 2.5: Künstlerische Darstellung des ANTARES Neutrinoteleskops [1]

zwei Effekte zu beobachten, zum einen Biolumineszenz, zum anderen der Zerfall des radioaktiven Kaliumisotops ^{40}K [1]. Das Kaliumisotop ^{40}K kann über zwei verschiedene Kanäle zerfallen, über den β- Zerfall eines Neutrons und den Elektroneneinfang eines Protons:

$$^{40}K \rightarrow {}^{40}Ca + e^- + \overline{\nu}_e \tag{2.9}$$

$$^{40}K + e^- \rightarrow {}^{40}Ar + \nu_e + \gamma \tag{2.10}$$

Das bei Gleichung 2.9 frei werdende Elektron ist relativistisch und kann somit im Wasser Tscherenkovstrahlung erzeugen, welche von den Photomultipliern gemessen wird.

Um Biolumineszenz und andere kontinuierliche Störquellen auszublenden werden Schwellen und Trigger eingesetzt, welche zum Teil online entscheiden ob gemessene Signale auf Störquellen oder Tscherenkovstrahlung eines Myons zurückzuführen sind.

Die erste Hürde, die ein Signal überwinden muss, ist ein gewisser Schwellwert für die von den Photomultipliern erzeugten Photoelektronen. Erst ab $0,3$ Photoelektronen wird das gemessene Signal eines OMs aufgezeichnet [9]. Dies bezeichnet man als „L0 Hit“.

Die nächste Selektion findet an Land statt. Hier wird festgestellt ob ein L0 Hit einen Schwellwert von 3 Photoelektronen übersteigt, oder ob es in einem Zeitfenster von $\pm 20\,ns$ übereinstimmende L0 Hits anderer OMs am selben Storey gibt. Wird eines der beiden Kriterien erfüllt, so bezeichnet man das Ereignis als „L1 Hit“.

Anschließend werden Triggeralgorithmen benutzt um kausal verknüpfte Hits zu finden. Es können mehrere Trigger, welche sehr unterschiedliche Algorithmen benutzen, parallel eingesetzt werden. Im Folgenden wird die Funktionsweise von zwei Triggern kurz beschrieben. Der erste Trigger stellt als Bedingung, dass mindestens fünf L1 Hits registriert wurden, welche so verteilt sind, dass sie der Spur eines Myons entsprechen könnten. Der zweite Trigger fordert mindestens zwei L1 Hits in drei aufeinander folgenden Storeys (innerhalb einer Line) in einem bestimmten Zeitfenster. Liegen die Storeys direkt nebeneinander, so beträgt dieses Zeitfenster $100\,ns$. Liegt ein Storey dazwischen, so ist das Zeitfenster $200\,ns$ groß. Spricht nun ein Trigger an, so werden alle Daten aller OMs innerhalb eines Zeitfensters von $\pm 2,2\,\mu s$ um den Detektionszeitpunkt des auslösenden Signals gespeichert und können weiter verarbeitet werden. Diese Daten werden dann als „Event“ oder Ereignis bezeichnet. Anhand dieser Daten werden anschließend weitere Informationen zu dem Teilchen berechnet. Hierzu zählen unter anderem die Spurrekonstruktion, die Energieschätzung und Informationen zu evtl. aufgetretenen Teilchenschauern.

2.4 Quellen hochenergetischer Teilchen

Es gibt verschiedene Prozesse, die als Quellen hochenergetischer Teilchen in Frage kommen und somit für ANTARES von Bedeutung sind.

2.4.1 Atmosphärische Myonen

Um Informationen über die Herkunft der Neutrinos zu erhalten misst man Myonen, welche aus charged current (CC) Reaktionen der Myonneutrinos mit anderen Teilchen entstanden sind. Das größte Hindernis bei dieser Methode sind relativistische Myonen. Solche Myonen entstehen in den oberen Atmosphärenschichten durch Reaktionen, welche u.a. von der kosmischen Strahlung hervorgerufen werden. Hochenergetische atmosphärische Myonen und Neutrinos entstehen hierbei meist durch den Zerfall von geladenen Pionen, welche beim Zusammenstoß von hochenergetischen Protonen der kosmischen Strahlung mit Teilchen innerhalb der Atmosphäre entstehen (siehe 2.4.2).

Die durchschnittliche Lebenszeit von Myonen beträgt nur $\tau_\mu \approx 2,2 \cdot 10^6\,\mu s$ [7]. Aufgrund der hohen Geschwindigkeiten und der damit verbundenen Zeitdilatation beträgt der durchschnittliche Fluss auf Höhe des Meeresspiegels aber immer noch ca. $100\,m^{-2}s^{-1}$. Myonen können zwar relativ ungehindert durch Luft fliegen, werden aber von Erde und Fels nahezu vollständig absorbiert. Um den Einfluss atmosphärischer Myonen auf die eigentliche Messung möglichst gering zu halten wurde das Teleskop so ausgerichtet, dass es sensitiver für Teilchen ist, die von unten nach oben fliegen. Somit wird die Masse der Erde als Teilchenfilter verwendet. Neutrinos hingegen können bis zu Energien von einigen TeV auch diese Barriere überwinden. Mit zunehmender Energie steigt aber auch der Wirkungsquerschnitt der Neutrinos, womit die Wahrscheinlichkeit, dass diese Teilchen das detektierte Volumen erreichen abnimmt.

Das Prinzip ist es also Myonen zu detektieren, die durch Reaktionen von nach oben fliegenden Neutrinos entstanden. Der Ort dieser Reaktion darf sich nicht zu weit entfernt vom Detektor befinden, da das Myon sonst vor dem Detektor noch durch andere Materie absorbiert wird (Abb. 2.2).

2.4.2 Erzeugungsmechanismen von Neutrinos

Das folgende Kapitel geht kurz auf mögliche Erzeugungsmechanismen von kosmischen Neutrinos ein.

Eine mögliche Reaktion bei der Neutrinos entstehen können, ist der Zusammen-

stoß eines Protons (p) mit einem Atomkern (A), siehe [4]:

$$p + A \rightarrow \pi^{\pm,0} + \text{weitere Hadronen} \tag{2.11}$$

$$\pi^0 \rightarrow \gamma + \gamma \tag{2.12}$$

$$\pi^{\pm} \rightarrow \mu^{\pm} + \overset{(-)}{\nu_{\mu}} \tag{2.13}$$

$$\mu^{\pm} \rightarrow e^{\pm} + \overset{(-)}{\nu_{\mu}} + \overset{(-)}{\nu_{e}} \tag{2.14}$$

Bei der Reaktion eines Protons mit einem Atomkern können neben anderen hadronischen Teilchen auch Pionen entstehen. Da Pionen selbst keine stabilen Teilchen sind ($\tau_{\pi^{\pm}} \approx 26\,ns$, $\tau_{\pi^0} \approx 8,4 \cdot 10^{-17}\,s$), zerfallen diese kurz nach dem Entstehen wieder. Der Zerfall von geladenen Pionen spielt vermutlich eine wichtige Rolle bei der Erzeugung von astronomischen hochenergetischen Neutrinos [14]. Bei der Reaktion 2.13, welche mit einer Wahrscheinlichkeit von $99,99\,\%$ den häufigsten Zerfall des geladenen Pions darstellt, wird ein Myon und ein (Anti) Neutrino erzeugt. Da das Myon ebenfalls kein stabiles Teilchen ist, zerfällt dieses nach kurzer Zeit wieder, wobei ebenfalls (Anti) Neutrinos entstehen (vgl. Reaktion 2.14) [7]. Neben den geladenen $\pi^{\pm}$ entstehen auch ungeladene π^0 welche zu zwei hochenergetischen Gammaquanten zerstrahlen. Somit können Quellen von hochenergetischen Neutrinos auch Quellen von hochenergetischer Gammastrahlung sein.

Bei einer weiteren Möglichkeit Pionen und damit Neutrinos zu erzeugen reagiert ein Proton mit einem hochenergetischen Photon und erzeugt hierbei ein Δ^+ Baryon:

$$p + \gamma \rightarrow \Delta^+ \rightarrow p + \pi^0 \tag{2.15}$$

$$p + \gamma \rightarrow \Delta^+ \rightarrow n + \pi^+ \tag{2.16}$$

Hierbei steht das n für ein Neutron.

Bei dem nachfolgenden Mechanismus reagieren Protonen mit der kosmischen Hintergrundstrahlung. Die für die Reaktion nötige Energie wird in diesem Fall von den Protonen bereitgestellt. Die Reaktionsgleichung ist äquivalent zu Gleichung 2.16 . Die Schwellenenergie, die ein Proton für eine solche Reaktion benötigt, hängt von dem an der Reaktion teilnehmenden Photon ab. Für Photonen aus der kosmischen Hintergrundstrahlung ist die Energieschwelle $E_p > 6 \cdot 10^{19}\,eV$. Stammt das Photon aus dem Infrarothintergrund, so beträgt die Energieschwelle $E > 10^{14}\,eV$ [4]. Solche hochenergetische Protonen treten unter anderem in kosmischer Strahlung auf.

2.4.3 Mögliche Neutrinoquellen

Nachdem auf das Funktionsprinzip und kurz auf mögliche Erzeugungsmechanismen für Neutrinos eingegangen wurde, soll eine kurze Übersicht zu möglichen

kosmischen Neutrinoquellen dargestellt werden. Für mehr Informationen zu den angerissenen Themen können die entsprechenden Quellen herangezogen werden.

Kosmische Strahlung Als kosmische Strahlung bezeichnet man aus dem Kosmos stammende Teilchen, die die Erde erreichen. Ein Großteil dieser Strahlung besteht aus Protonen, es sind aber auch schwerere Atomkerne, Elektronen, Photonen und Neutrinos enthalten [6]. Teilchen der kosmischen Strahlung können Energien von bis zu $10^{20}\,eV$ erreichen [14].

Da kosmische Strahlung zu großen Teilen aus geladenen Teilchen besteht, ist es nur schwer möglich ihren Ursprung zu bestimmen, da die Flugbahn der Teilchen von kosmischen Magnetfeldern immer wieder beeinflusst wird. Aufgrund der zum Teil sehr hohen Energien der Protonen können diese Neutrinos erzeugen (siehe Abschnitt 2.4.2). Da Neutrinos nicht geladen sind, wird ihre Flugbahn nicht von magnetischen Feldern beeinflusst. Die Detektion dieser Neutrinos könnte also bei der Identifizierung der Ursprünge kosmischer Strahlung entscheidende Hinweise liefern.

AGN Als aktive galaktische Kerne (engl. Active Galactic Nuclei, AGN) bezeichnet man extrem helle Zentren von Galaxien, die manchmal um mehrere Größenordnungen heller als der Rest der Galaxie leuchten. Abhängig unter anderem vom Spektrum der Strahlung werden Galaxien mit einem aktiven Kern noch weiter unterteilt (siehe [6, 8, 12]). Großes Interesse genießen hierbei sog. Quasare (engl. quasi stellar object), welche in Anbetracht der sehr großen Entfernung immer noch ungewöhnlich hell leuchten.

Pulsarwindnebel An den Polen des magnetischen Dipolfeldes eines Pulsars können hochrelativistische Teilchen emittiert werden. Diese Teilchen bilden eine den Pulsar umgebende Wolke, was als Pulsarwindnebel bezeichnet wird. Aufgrund der hohen Energien, die die Teilchen besitzen, können Reaktionen zur Erzeugung von hochenergetischen Neutrinos stattfinden. Da Neutrinos ungeladen sind, werden sie von dem starken magnetischen Feld eines Pulsars nicht abgelenkt und können dessen Einflussbereich verlassen [14].

Supernovaüberreste Bei einer Supernova wird unter anderem die äußere Gashülle des explodierenden Sterns mit Geschwindigkeiten von bis zu $10000\,km/s$ abgestoßen [6]. Trifft diese abgestoßene Hülle auf ein umgebendes Medium, so entsteht eine Schockwelle. Innerhalb dieser Schockwelle können Teilchen auf relativistische Geschwindigkeiten beschleunigt werden [14]. Hochenergetische hadronische Teilchen können dann die in Abschnitt 2.4.2 beschriebenen Reaktionen durchlaufen und hochenergetische Neutrinos erzeugen.

Gammastrahlenblitze (GRB) Gammastrahlenblitze (engl. gamma ray bursts, GRB) sind kurz aufleuchtende Pulse im Bereich der Gammastrahlung ($f < 10^{19}\,Hz$). Meist ist nach den kurzen Pulsen im kurzwelligen Gammabereich auch Strahlung im langwelligeren Bereich zu sehen, welche nach und nach abklingt (sog. „afterglows“). Der Ursprung von GRBs ist noch nicht vollständig ergründet. Eine mögliche Quelle können Hypernovae sein, Supernovae die ungewöhnlich hell erscheinen. Eine weitere Theorie erklärt extrem kurz ($< 2\,s$) aufleuchtende GRBs mit dem Verschmelzen von Doppelsternsysteme, die aus zwei Neutronensternen oder einem schwarzem Loch und einem Neutronenstern entstehen [6].

Kapitel 3

Random Decision Forests

In Kapitel 2.4.1 wurde bereits das Problem der atmosphärischen Myonen und die Störung bei der Messungen von kosmischen Neutrinos durch diese erläutert. In dem folgenden Kapitel soll nun der theoretische Hintergrund zu einer vielversprechenden Lösung dieses Problems erläutert werden. Hierbei handelt es sich um „Random Decision Forests“ (RDFs). RDFs entstammen einem Teilgebiet der Informatik, namentlich der Mustererkennung, und sind eine Weiterentwicklung der vielfach genutzten Entscheidungsbäume. Der Kerngedanke ist es die verschiedenen Ereignisse, die von einem Neutrinodetektor registriert werden können, von einem für die Mustererkennung geschriebenen Algorithmus in vorher definierte Klassen, wie etwa „von oben kommendes Teilchen“ und „von unten kommendes Teilchen“, einteilen zu lassen.

3.1 Mustererkennung

Mustererkennung ist nicht nur ein Begriff aus der Informatik, sonder ganz allgemein gesehen der Vorgang in einer Menge an Informationen Regelmäßigkeiten und Gesetzmäßigkeiten zu erkennen. Somit spielt die Mustererkennung für die menschliche Wahrnehmung ebenfalls eine entscheidende Rolle.

Funktionsprinzip Das Ziel eines Klassifikators ist es einer Menge an Informationen, im Folgenden als Datensatz bezeichnet, eine Klasse zuzuordnen. Dies wird mittels bestimmter Eigenschaften, die ein Datensatz besitzt, bewerkstelligt. Geht es zum Beispiel um die Unterscheidung zwischen zwei verschiedenen Obstsorten, so könnte eine Klasse die Bezeichnung „Apfel“ und eine andere die Bezeichnung „Melone“ tragen. Die Menge an Informationen, also der Datensatz, mit dem diese Unterscheidung erfolgt, hängt vom Klassifikator und der Klassifikationsaufgabe ab. In diesem Beispiel soll der Klassifikator ein Mensch sein. Die Informationen, die

man über ein zu klassifizierendes Objekt erhält, können zum Beispiel die Farbe, der Geruch, die Konsistenz und das Gewicht sein. Nachdem man von dem Objekt die entsprechenden Informationen erhalten hat, sollte die Einteilung in eine der beiden Klassen mit Hilfe dieser Informationen relativ sicher gelingen.

Der Klassifikator, der für diese Arbeit verwendet wurde, ist ein numerischer Algorithmus, weshalb die Informationen, die von einem Datensatz gewonnen werden können, letztendlich in Form von Zahlen vorhanden sein müssen. Um bei dem letzten Beispiel zu bleiben könnte man also für die Farbe den RGB Wert angeben und das Gewicht in kg. Diese den zu klassifizierenden Objekten entnommenen Informationen werden allgemein als Kennzahlen bezeichnet. Zusammenfassend ist das Prinzip einer Klassifikation also das Extrahieren von Kennzahlen aus Datensätzen und anhand dieser die Einteilung in Klassen.

Die Anwendung der Mustererkennung in der Informatik erfolgt in vielen Fällen nach den im Folgenden beschriebenen Schritten [10].

Datenerfassung Zunächst müssen Daten, welche bearbeitet werden sollen, erfasst werden. Dies geschieht meist durch unterschiedlichste Sensoren, wie z.B. Mikrophone für akustische Signale und Kameras für visuelle Signale. Nach dem Erfassen, z.B. eines akustischen Signals, muss dieses, um weiter verarbeitet werden zu können, zunächst noch digitalisiert und hierzu diskretisiert werden. Die Art und Weise wie dieser Schritt bewerkstelligt wird, kann maßgeblich die Erfolgsrate nachfolgender Schritte beeinflussen (mehr hierzu in [5]).

Die Daten, für welche der in dieser Arbeit verwendete Algorithmus geschrieben wurde, sind mit dem ANTARES Neutrinodetektor erfasst worden. Die Digitalisierung findet online und vor Ort statt (siehe 2.3). Für die Evaluation eines Klassifizierungsalgorithmus müssen Simulationen verwendet werden, da bei echten Daten vorher nicht bekannt ist, um welche Typen von Events es sich handelt.

Vorverarbeitung In vielen Fällen wird ein Datensatz, bevor die eigentlichen Schritte der Mustererkennung statt finden, vorverarbeitet. Hierbei geht es darum, unnötige und störende Anteile innerhalb der Daten herauszufiltern, ohne dabei Informationen zu verlieren. Ein einfaches und einleuchtendes Beispiel wäre das Herausfiltern von Hintergrundgeräuschen bei einem Telefonat.

Extraktion der Kennzahlen Die erste große Hürde, die ein erfolgreicher Algorithmus zur Klassifikation überwinden muss, ist das Bestimmen von aussagekräftigen Kennzahlen (engl. Features), da alle nachfolgenden Schritte, inklusive der abschließenden Klassifizierung anhand dieser Kennzahlen vorgenommen werden. Werden also für eine Klassifizierung unpassende Kennzahlen gewählt, so ist es kaum mehr möglich gute Ergebnisse mit einem beliebigen Algorithmus zu erzielen.

Leider ist es nicht immer von vorn herein möglich die Kennzahlen, die den größten Informationsgehalt enthalten, sofort zu identifizieren. Je komplexer die Datensätze und die Klassen werden, desto schwieriger ist die Auswahl sinnvoller Kennzahlen. Sobald ein Satz sinnvoller Kennzahlen festgelegt ist, müssen diese für den zu klassifizierenden Datensatz berechnet werden. Die Menge der Kennzahlen und der Aufwand diese zu berechnen beeinflusst offensichtlich die Laufzeit des Algorithmus.

Es ist außerdem interessant zu erwähnen, dass eine größere Anzahl von Kennzahlen nicht zwingend ein besseres Ergebnis bei der Klassifizierung liefert. Überraschender Weise ist bei vielen Klassifikatoren sogar eher das Gegenteil der Fall [5] (Beweis: [2]).

Selektion der Kennzahlen Bei den meisten Klassifikationsroutinen wird nach der Extraktion der Kennzahlen ein weiterer Algorithmus verwendet um aus der Menge aller vorhandenen Kennzahlen eine kleinere Menge von optimalen Kennzahlen zu finden. Grund hierfür ist das eben erwähnte Problem, dass sehr viele Kennzahlen oft ein eher schlechteres Ergebnisse liefern. Ein weiterer Vorteil einer kleineren Menge an Kennzahlen ist der Performancegewinn, welcher bei online Algorithmen eine entscheidende Rolle spielt. Es gibt viele verschiedene Möglichkeiten eine Menge an optimalen Kennzahlen zu finden [10]. In Abschnitt 4.4 wird die in dieser Arbeit verwendete Methode genau erläutert.

Training In diesem Schritt wird nun dem Klassifikator beigebracht, wie er anhand der berechneten Features Datensätze in einzelne Klassen unterteilt. Es gibt hier prinzipiell zwei unterschiedliche Methoden, zum einen das „supervised learning“, bei dem die Klassen genau definiert sind, und zum anderen das „unsupervised learning“, bei dem der Algorithmus selbst entscheidet welche Eigenschaften eine Klasse hat und teilweise auch wie viele es gibt. Bei dem in dieser Arbeit verwendeten Klassifikator, dem RDF, sind vor dem Trainieren die Klassen genau definiert. Es handelt sich somit um „supervised learning“. Bei vielen Algorithmen, die zur Mustererkennung verwendet werden, gibt es neben den interessanten Klassen meist noch eine „Rejection Class“. Hierbei handelt es sich um eine Klasse in die Datensätze eingeordnet werden, deren reguläre Ergebnisse nicht eindeutig sind oder bei denen die Sicherheit mit der eine Zuordnung in eine Klasse erfolgt zu gering ist. Der Grund hierfür ist die globale Sicherheit mit der klassifiziert wird möglichst hoch zu halten.

Um nun einen Klassifikator zu Trainieren (supervised learning) benötigt man Beispiele der einzelnen Klassen, also Datensätze, die bereits die Information enthalten zu welcher Klasse sie gehören. Handelt es sich zum Beispiel um einen Algorithmus der handschriftlich verfasste Dokumente verarbeiten soll, so muss der Algorithmus wissen, wie per Hand geschriebene Buchstaben aussehen können.

Klassifizierung Der letzte Schritt bewerkstelligt nun die Einteilung von Daten in vorher definierte Klassen. Hierzu wird ein Satz Kennzahlen aus den zu klassifizierenden Daten berechnet und mittels des trainierten Algorithmus einer Klasse zugeordnet.

Möchte man die Genauigkeit mit der ein Klassifikator arbeitet bewerten, so müssen in diesem Schritt Daten mit bekannter Zuordnung verwendet werden. Anhand der Ergebnisse des Klassifikators für jene Daten lässt sich dann eine Sicherheit für den Algorithmus berechnen (mehr zur Evaluierung in 4.3).

3.2 Decision Trees und RDFs

Nachdem im letzten Abschnitt die Idee der Klassifizierung kurz erklärt und einige Schritte oberflächlich erläutert wurden, soll im Folgenden auf eine mögliche Umsetzung dieser Prinzipien eingegangen werden. Es handelt sich hierbei um „Decision Trees“ und die auf diesen aufbauenden RDFs.

3.2.1 Decision Tree

Entscheidungsbäume stellen eine einfache und intuitive Möglichkeit zur Klassifizierung von Daten dar und werden deshalb auch in vielen anderen Bereichen außerhalb der Wissenschaft zu Hilfe genommen.

Die Funktionsweise eines Entscheidungsbaumes lässt sich sehr anschaulich mit Hilfe eines Baumdiagramms darstellen (Abb: 3.1).

Bei diesem Entscheidungsbaum wird anhand der zwei Kennzahlen ($F1$ und $F2$) in zwei Klassen ($C1$ und $C2$) unterteilt. Der Entscheidungsbaum wird von oben nach unten Schritt für Schritt abgearbeitet. Die erste Entscheidung die zu treffen ist, hängt von $F1$ ab. Für Werte kleiner als $0,5$ wird dem linken Ast gefolgt, ansonsten dem rechten. Alle weiteren Entscheidungen, in einem Baumdiagramm auch Knoten genannt, werden nach dem selben Prinzip getroffen. Nach einer bestimmten Anzahl an Knoten gelangt man schließlich an das Ende des Baumes, welches die zugeteilte Klasse bestimmt. In Abbildung 3.1 sind diese (sog. „Leaves“) durch Rechtecke dargestellt, die den Klassennamen enthalten.

Im allgemeinen ist ein Entscheidungsbaum wesentlich größer und komplexer aufgebaut um sinnvolle Ergebnisse zu liefern. Zunächst kann ein Knoten an mehr als zwei Äste angebunden sein und die Entscheidungsregel muss nicht zwangsläufig mit einem Schwellwert zusammenhängen, sondern kann mittels einer Funktion, die von einer oder mehreren Kennzahlen abhängig ist, ermittelt werden. Des weiteren werden meist mehr als zwei Kennzahlen für einen Datensatz berechnet um diesen zu klassifizieren. Die Einteilung in eine Klasse ist in vielen Fällen mit einer Wahrscheinlichkeit verknüpft. Erreicht ein Algorithmus also den letzten Ast ei-

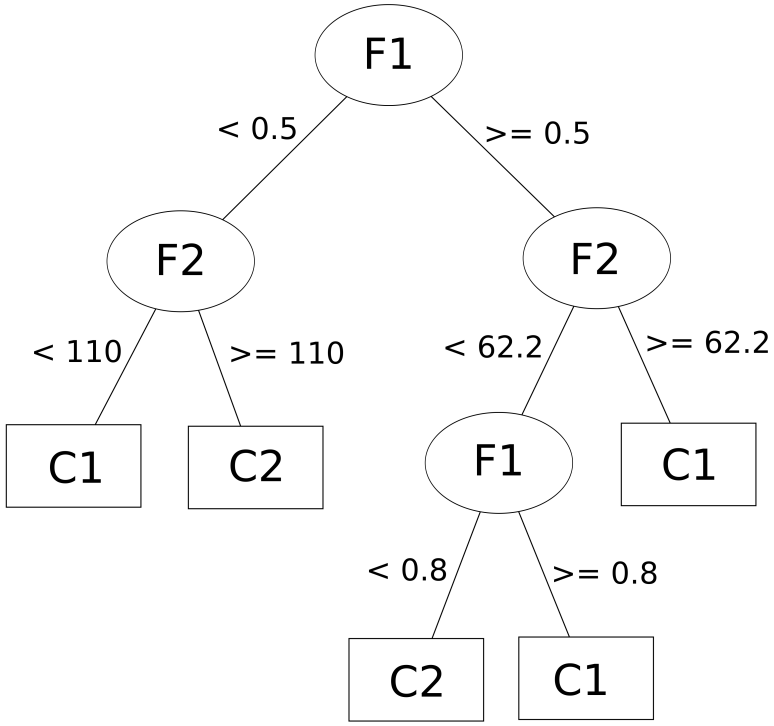

Abbildung 3.1: Schematische Darstellung eines binären Entscheidungsbaumes [10]

nes Entscheidungsbaumes, so kann noch eine Wahrscheinlichkeit, die die Sicherheit des Algorithmus was die Einteilung der Daten in diese Klasse angeht, angegeben werden. Diese Wahrscheinlichkeit kann für die einzelnen Leaves statisch sein, oder aber abhängig von den vorher getroffenen Entscheidungen, für jeden Fall einzeln berechnet werden [5].

3.2.2 Random Decision Forests

Entscheidungsbäume genießen eine große Beliebtheit bei Klassifizierungsproblemen, da die Idee hinter diesen sehr intuitiv ist und sie vergleichsweise wenig Rechenzeit benötigen. Dennoch gibt es einen großen Nachteil, der einfache Entscheidungsbäume bei komplexeren Problemen untauglich macht. Es handelt sich hierbei um die nur begrenzt vorhandene Fähigkeit die beim Training erhaltenen Informationen zu generalisieren. Das bedeutet, dass trainierte Entscheidungsbäume, wenn diese unbekannte Daten klassifizieren sollen, den Hang zu einer bestimmten Klasse besitzen und vor allem schlechtere Ergebnisse liefern [11]. Selbst wenn die Daten, die zum Trainieren benutzt wurden, nach dem Training zu 100% richtig klassifiziert werden können, so ist diese Tendenz bei anderen Daten zu erkennen.

Eine Methode die Tragweite dieses Problems zu verringern sind die „Random Decision Forests“. Die Idee bei RDFs ist es, nicht nur einen sondern viele einfache Entscheidungsbäume zu trainieren, um nicht der Tendenz eines einzelnen zu unterliegen. Hierbei werden die einzelnen Entscheidungsbäume aber nicht mit dem gesamten Kennzahlenvektor trainiert, sondern mit einem beliebigen kleineren Teilvektor. Für einen $m-$dimensionalen Kennzahlenraum gibt es hierfür 2^m Möglichkeiten, was bei hochdimensionalen Kennzahlenräumen den Bereich des Möglichen deutlich übersteigt. Welche Teilmengen der maximal zur Verfügung stehenden Kennzahlen letztendlich gewählt werden, um die einzelnen Bäume zu trainieren, wird per Zufall entschieden. Außerdem werden beim Training der einzelnen Entscheidungsbäume nicht alle zur Verfügung stehenden Trainingsdaten hergenommen, sondern ebenfalls nur eine Teilmenge. Diese beiden Prinzipien der RDFs sorgen dafür, dass die einzelnen Entscheidungsbäume unterschiedliche Tendenzen bezüglich bestimmten Klassen entwickeln, was bei einer großen Menge an Entscheidungsbäumen dann dazu führt, dass RDFs nicht dem Hang zu einer bestimmte Klasse unterliegen.

Experimente haben gezeigt, dass RDFs auf unbekannten Daten deutlich bessere Ergebnisse liefern als herkömmliche Entscheidungsbäume, was dafür spricht, dass die Fähigkeit von den Daten, mit denen trainiert wurde, auf unbekannte Daten zu generalisieren ausgeprägter ist. Es wurde außerdem mehrfach festgestellt, dass die Sicherheit bei der Klassifizierung mit der Anzahl der verwendeten Bäume nicht abnimmt. Dies ist eine bemerkenswerte Eigenschaft von RDFs, da bei den meisten anderen Klassifikatoren die Klassifizierungssicherheit mit zunehmender Komplexität abnimmt [11] [10].

Kapitel 4

Implementierung

Das Kernthema dieser Arbeit ist die Erweiterung des bisher verwendeten Kennzahlenvektors und die Überprüfung, ob sich durch diese Veränderung die Klassifizierungssicherheit für verschiedene Klassifikationsprobleme signifikant verbessert. In dieser Arbeit wurde die Klassifizierung in „von oben kommende Teilchen“ und „von unten kommende Teilchen“ (kurz UpDown Klassifizierung) sowie die Klassifizierung in verschiedene Energiebereiche untersucht.

Im folgenden Abschnitt wird die hauptsächlich verwendete Software, das Programm „RDFClassify“ (entwickelt von Stefan Geißelsöder, 2011, [10]), beschrieben. Außerdem wird erläutert, wie der bereits vorhandene Code erweitert wurde, um entsprechende Daten zur Überprüfung der Kernfrage dieser Arbeit zu erlangen. Die für die Auswertung verwendete Software wird ebenfalls ausführlich erklärt.

In Abschnitt 4.4 dieses Kapitels wird beschrieben, wie aus der Menge aller Kennzahlen eine deutlich kleinere Menge selektiert wird, welche die wirkungsvollsten Kennzahlen enthält.

4.1 Aufbau und Anwendung der Software „RDFClassify“

Die im Seatray Framework eingebettete Software RDFClassify ist ein Klassifikator, welcher mit Hilfe der „Random Decision Forests“ von ANTARES aufgezeichnete Events in vorher definierte Klassen einteilt.

RDFClassify besteht aus zwei Modulen, „I3RDFFeatureExtract“ und „I3RDFClassify“. Damit ein Klassifikator funktioniert müssen die in Kapitel 3.1 beschriebenen Schritte abgearbeitet werden.

Die Datenerfassung ist an dem Zeitpunkt, an dem ein Klassifikator beginnt zu arbeiten, bereits abgeschlossen. Wie bereits erwähnt, findet die Datenerfassung

durch das ANTARES Neutrinoteleskop statt, bzw. durch die entsprechenden Simulationen.

Der nächste Schritt ist die Vorverarbeitung der Daten. Für die von ANTARES aufgezeichneten Daten und entsprechende Simulationen zählt hierzu das Filtern und Triggern (siehe Abschnitt 2.3), was von extra hierfür entwickelten Modulen übernommen wird. Die Vorverarbeitung von Daten und Simulationen wird also durch andere Module bewerkstelligt, bevor diese an RDFClassify übergeben werden.

Die Extraktion der Kennzahlen ist der erste Schritt, der von RDFClassify übernommen wird. Hierfür steht das Modul I3RDFFeatureExtract zur Verfügung. I3RDFFeatureExtract kann auf zwei verschiedene Weisen verwendet werden. Zum einen lediglich zu dem Zweck, um aus einem Datensatz bestimmte Kennzahlen zu berechnen, und zum anderen um diesem Datensatz eine Klasse zuzuordnen (intern werden die gleichen Kennzahlen verwendet, nur wird bei ersterem nicht die Nummer der Klasse mit ausgegeben, sondern die Zahl 99, welche für „keiner Klasse zugeordnet“ steht). Letztere Methode ist beim Trainieren eines RDF nötig, da der Klassifikator hierfür wissen muss, welcher Klasse ein Datensatz angehört.

Der nächste Schritt ist das Trainieren. Diese Aufgabe wird von dem Modul I3RDFClassify übernommen. Hierfür benötigt das Modul Datensätze, zu denen alle Kennzahlen bereits berechnet wurden und die bereits zu einer bestimmten Klasse zugeteilt wurden. Im allgemeinen gilt, dass ein Training mit mehr Trainingsdaten zu einem Klassifikator führt, welcher bessere Ergebnisse liefert. Wird im Laufe einer Auswertung der Kennzahlenvektor oder die Eigenschaften der Klassen verändert, so muss auch der RDF mit entsprechend angepassten Trainingsdaten neu trainiert werden, um weiterhin vertrauenswürdige Ergebnisse liefern zu können. Ein Kennzahlenvektor kann zum Beispiel durch Hinzufügen oder Entfernen von Kennzahlen verändert werden. Außerdem können sich die Berechnungen der Kennzahlen ändern. Ein trainierter RDF wird von dem Modul in eine extra Datei mit der Endung „.rdf“ gespeichert. Der Ort dieser Datei muss dem Klassifikator zum späteren Klassifizieren übergeben werden.

Der letzte Schritt beinhaltet die Zuordnung von Ereignissen in eine bestimmte Klasse. Dieser Schritt wird von dem Module I3RDFClassify bewerkstelligt. Damit das Modul entsprechend arbeiten kann benötigt es Zugriff auf die bereits extrahierten Kennzahlen der zu klassifizierenden Daten. Des weiteren muss man den Ort des mit passenden (anderen) Daten trainierten RDFs übergeben. Wird ein RDF übergeben, welcher mit unpassenden Daten trainiert wurde, also passen z.B. die Größen der Kennzahlenvektoren nicht zusammen, so wird das Programm in aller Regel mit einer Fehlermeldung beendet. Der Klassifikator teilt nun jeden Datensatz in eine bestimmte Klasse ein und gibt entsprechende Informationen an verschiedenen Stellen aus. Neben der zugeteilten Klasse wird noch eine sog. „RDFSafety“ berechnet.

Dieser Wert ist nicht mit der Klassifizierungssicherheit des trainierten RDFs zu verwechseln. Die RDFSafety gibt lediglich den Prozentsatz an, mit dem sich die einzelnen Entscheidungsbäume innerhalb des RDFs bei der letztendlich getroffenen Einteilung einig waren. Die allgemeine Klassifizierungssicherheit eines trainierten RDFs muss anders berechnet werden (siehe 4.3).

4.2 Zusätzliche Kennzahlen

Wie bereits erwähnt, ist die Hauptfragestellung die Erweiterung des bisher genutzten Kennzahlenvektors mit einer Länge von 143 Kennzahlen und die Bestimmung der Klassifizierungssicherheit für verschiedene Klassifikationsprobleme. Die Motivation hinter dieser Fragestellung ist die intuitive Idee, dass ein Klassifikator, welcher mit aussagekräftigen Kennzahlen arbeitet, deutlich bessere Ergebnisse liefern kann. Die in Abschnitt 3.1 erwähnte Verschlechterung der Klassifizierungssicherheit beim Hinzufügen von zusätzlichen schlechten Kennzahlen kann zwar auch bei RDFs auftreten, diese sind aber im Vergleich zu vielen anderen Klassifikatoren sehr resistent gegen schlechte Kennzahlen, solange diese nicht einen Großteil der insgesamt genutzten Kennzahlen ausmachen.

Ein weiterer Grund die neue Kennzahlen zu erschließen und damit die Menge der zur Verfügung stehenden Kennzahlen zu erhöhen ist die Idee den Kennzahlenvektor für verschiedene Klassifikationen zu optimieren. So erscheint es zum Beispiel logisch, für die Klassifizierung in „von oben kommende Teilchen“ und „von unten kommende Teilchen“ die Ergebnisse einer Spurrekonstruktion als Kennzahlen zu verwenden.

4.2.1 Quellen weiterer Kennzahlen

Wie bereits in Abschnitt 2.3 erwähnt werden bei einem als Event getriggertem Ereignis alle Informationen aller OMs innerhalb eines $4,4\,\mu s$ großen Zeitfensters gespeichert. Für jedes OM sind diese Informationen dann in Form von vierdimensionalen Vektoren verfügbar. Diese Vektoren enthalten die folgenden Informationen:

- Zeitpunkt
- Nummer der Line
- Nummer des OM
- Ladung

139 der 143 bisher verwendeten Kennzahlen werden in dem Modul I3RDFFeatureExtract anhand dieser Daten berechnet. Hierbei handelt es sich zum Beispiel um Kennzahlen wie die größte Ladung eines Storeys innerhalb des gesamten

Events oder die Etagennummer des Storeys mit maximaler Ladung (eine vollständige Liste der Kennzahlen ist in Tabelle B.1 zu finden).

Die nächsten vier Kennzahlen stammen aus dem Modul „AAFit“, welches zur Spurrekonstruktion geschrieben wurde. Bei den verwendeten Kennzahlen handelt es sich um je einen rekonstruierten Zenit- und Azimutwert, sowie zwei Qualifikationsparameter, welche die Güte der Spurrekonstruktion beziffern.

Der Kennzahlenvektor aus diesen 143 Kennzahlen stellt den bisherigen Standardkennzahlenvektor dar, der in dieser Arbeit erweitert werden soll. Die Kennzahlen, die zur Erweiterung dieses Vektors verwendet wurden, stammen aus drei weiteren Modulen, welche jeweils für andere Rekonstruktionen geschrieben wurden. Es handelt sich hierbei um die Module „AntEnergyReco“, ein Modul zur Rekonstruktion der Teilchenenergie, „DusjShowerReco“, welches zur Rekonstruktion von hadronischen oder elektromagnetischen Schauern innerhalb des Detektors verwendet wird und „rrFitReco“, was eine Erweiterung des Moduls AAFit darstellt.

Grund für die Wahl von Kennzahlen gerade aus diesen Modulen sind die Klassifikationen, für die die neuen Kennzahlenvektoren getestet werden sollen. Zum einen soll weiterhin nach „von oben kommende Teilchen “ und „von unten kommende Teilchen“ (kurz UpDown Klassifikation) klassifiziert werden, weshalb sich Kennzahlen aus Spurrekonstruktionen anbieten.

Eine weitere Klassifikation, welche so vorher noch nicht erprobt wurde, soll Events in verschiedene Energieklassen einteilen, was die Motivation zur Verwendung von Kennzahlen aus dem Modul zur Energierekonstruktion liefert.

In [10] wurde bereits ein Klassifikator trainiert, welcher in die Klassen „reine Teilchenschauer“ und „Teilchenspur“ trennen soll, welcher also entscheiden soll, ob es sich bei einem Event nur um Teilchenschauer handelt, oder ob es sich um eine Teilchenspur mit evtl. vorhandenen Schauerereignissen handelt. Diese Form der Klassifizierung wurde im Rahmen dieser Arbeit nicht weiter untersucht, dennoch wurden Kennzahlen aus dem Modul DusjShowerReco mit dieser Motivation verwendet.

4.2.2 Implementierung der neuen Kennzahlen

Um nun die entsprechenden Kennzahlen aus den oben genannten Modulen verwenden zu können müssen diese immer, wenn Kennzahlen zu Datensätzen berechnet werden, ebenfalls aufgerufen und ausgeführt werden. Im Seatray Framework ist es üblich, die hierfür geschriebenen Module mit Hilfe von Pythonskripten aufzurufen. An ein solches Skript werden meist auch die zu bearbeitenden Daten übergeben. Bevor entsprechende Module aber übergebene Daten bearbeiten können, müssen diese noch entsprechend „aufbereitet werden“.

Die verwendeten simulierten Daten beinhalten keine gefilterten und getriggerten Hits von verschiedenen möglichen Ereignissen, sondern nur die Information,

wie viele Photonen zu welchem Zeitpunkt an welchem OM hätten registriert werden können, mit anderen Worten die größtmöglich registrierbare Intensität. Solche Daten sind für echte Messungen selbst bei sehr guten Bedingungen unmöglich.

Um nun die Simulationen näher an wirkliche Daten heranzurücken wird zunächst ein Hintergrundrauschen, welches u.a. Biolumineszenz und den Zerfall von radioaktiven Kalium imitiert, hinzugefügt. Als nächstes wird ein sog. „PMT- Simulator“ verwendet, welcher das charakteristische Messverhalten der Photomultiplier imitiert. Des weiteren wird jedem Skript, das mit Simulationen läuft, eine so genannte „Detector Map“ übergeben. Innerhalb dieser Map sind Informationen zum Zustand des Detektors enthalten, also angaben darüber wie viele OMs momentan funktionsfähig sind und welche Position die Lines gerade einnehmen. Im nächsten Schritt wird berechnet, wie viele Photoelektronen innerhalb der Integrationszeit von einem OM gemessen werden. Nachdem diese Effekte zu den ursprünglichen Simulationen hinzugefügt wurden, berechnen andere Module welche L0 und L1 Hits gemessen worden wären. Anschließend können verschiedene Trigger innerhalb dieser Daten Events markieren, welche dann gespeichert und an weitere Module übergeben werden.

Die meisten Module, deren Aufgabe es ist eine Rekonstruktion durchzuführen, verwenden die getriggerten Events, so auch die für die Extraktion der neuen Kennzahlen benötigten Module zur Schauer-, Energie- und Spurrekonstruktion. Die berechneten Größen dieser Module werden als I3Objekt (meist eine Abwandlung eines Standard C++ Objekts) in den entsprechenden Frame geschrieben und können innerhalb des für Seatray üblichen „.i3“ Datenformats gespeichert und abgerufen werden. Es ist aber auch möglich Informationen, die von einem Modul in einen Frame geschrieben werden, vom nächsten Modul, welches im gleichen Skript aufgerufen wird, abrufen zu lassen, was wesentlich effizienter ist als der Umweg über eine externe Datei.

Das Skript, welches zur Extraktion der Kennzahlen für diese Arbeit geschrieben wurde, hat also (sehr vereinfacht) die folgende Struktur:

- „Aufbereiten“ der Simulationen
- Aufruf der zusätzlichen Module (Schauer-, Energie-, Spurrekonstruktion)
- Aufruf des Moduls I3RDFFeatureExtract und Übergabe der neuen Kennzahlen an dieses

Der Quellcode des Moduls I3RDFFeatureExtract wurde soweit angepasst, dass die neuen Kennzahlen an den bereits implementierten Kennzahlenvektor angefügt und anschließend entsprechend gespeichert werden.

Um eine möglichst breit gefächerte Analyse zu ermöglichen wurden zusätzlich zu der Erweiterung des Kennzahlenvektors drei Schalter implementiert, mit welchen es möglich ist, zu steuern, welche Kennzahlen (von welchem Modul) an den

Standardkennzahlenvektor angehängt werden. Die Motivation für diesen Schritt ist es, alle Varianten eines erweiterten Kennzahlenvektor berechnen und mögliche Effekte auf Klassifizierungen möglichst detailliert untersuchen zu können.

4.2.3 Verwendete Datenformate und deren Inhalte

Das Modul I3RDFFeatureExtract wurde so geschrieben, dass es die extrahierten Kennzahlen standardmäßig in Form von „.dat“ Dateien abspeichert. Zusätzlich zu diesen wird bei jedem Aufruf noch eine „.meta“ Datei erzeugt, welche zusätzliche Informationen zu jedem Event enthält. In den „.meta“ Dateien stehen auch Informationen zu den Simulationen, also unter anderem Energie, Zenit und Azimut mit der ein entsprechendes Event simuliert wurde. Natürlich werden die Informationen, die in den „.meta“ Dateien stehen, nicht bei Klassifizierungen verwendet.

Nachdem I3RDFFeatureExtract entsprechend angepasst wurde und ein passendes Pythonskript, welches für die Berechnung aller Kennzahlen benutzt werden kann, erstellt wurde, fehlt noch ein Programm mit dem die erwünschte Vielfältigkeit der Klassifikationsexperimente erreicht wird. Mit Vielfältigkeit ist gemeint, dass jede Variante eines möglichen erweiterten Kennzahlenvektors für jede Klassifikation benutzt und ausgewertet werden kann. Theoretisch wäre es natürlich möglich wirklich jede Variante eines erweiterten Kennzahlenvektors zu testen. Da es sich bei 47 neuen Kennzahlen aber um 2^{47} Möglichkeiten handelt, wurden die Kennzahlen, die von einem Modul stammen, zu einem Block zusammengefasst und können nur gemeinsam hinzugefügt oder abgeschnitten werden. Dies führt zu acht verschiedenen Kennzahlenvektoren, welche für sechs verschiedene Klassifikationen getestet wurden. Bei diesen sechs Klassifikationen handelt es um fünf Klassifikationen nach Energieklassen und eine UpDown Klassifikation.

Die acht verschiedenen Kennzahlenvektoren werden in den folgenden Abschnitten mit einer dreistelligen Binärzahl referenziert. Jede Stelle dieser Zahlen steht für die Kennzahlen aus einem Modul, bzw. ob diese verwendet wurden oder nicht. Die Stellen stehen für die Module in der folgenden Form: AntEnergyReco - DusjShowerReco - rrFitReco. Somit würde also der Kennzahlenvektor mit der Bezeichnung 101 alle Kennzahlen aus dem Modul „AntEnergyReco“ enthalten, keine Kennzahlen aus „DusjShowerReco“ und alle Kennzahlen aus „rrFitReco“.

4.2.4 Das Programm „preProcess“

Wie bereits erwähnt kann I3RDFFeatureExtract neben der Berechnung von verschieden großen Kennzahlenvektoren auch Klassen zuweisen, was für das Training eines RDF nötig ist. Möchte man also für eine Klassifizierung zwei verschiedene Kennzahlenvektoren testen, so müsste I3RDFFeatureExtract zweimal für alle Daten die entsprechenden Kennzahlenvektoren berechnen. Da es sich aber nicht um

disjunkte Mengen von Kennzahlen handelt, sondern der kleinere Kennzahlenvektor vollständig in dem größerem enthalten ist, ist es rechentechnisch günstiger einmal den größeren der beiden Vektoren berechnen zu lassen und diesen für die Klassifizierung entsprechend zurechtzuschneiden.

Des weiteren ist es unvorteilhaft, für jede Klassifizierung jeden Kennzahlenvektor berechnen zu müssen, da die Kennzahlenvektoren nicht von den Klassifizierungen abhängen. Zu welcher Klasse ein Event zugehörig ist, ist ebenfalls in den „.meta“ Dateien gespeichert. Um nun mit dem gleichen Kennzahlenvektor mehrere Klassifizierungen zu testen, müssen also nur entsprechende Stellen in den „.meta“ Dateien geändert werden, was gegenüber dem neu Berechnen der Kennzahlen ein wesentlicher Performancegewinn ist.

Die eben beschriebenen Aufgaben können mit dem Programm „preProcess“ bewerkstelligt werden. Hierbei handelt es sich um ein Programm mit, dem man nach dem Berechnen der Kennzahlen die „.dat“ Dateien, in welchen die Kennzahlenvektoren gespeichert sind, und die „.meta“ Dateien, in welchen u.a. die Informationen zur Klassenzugehörigkeit und zu den Simulationen enthalten sind, gezielt bearbeiten kann.

Für diese Arbeit war es also nötig zum einen den Kennzahlenvektor zurecht zu schneiden und zum anderen Klassenzuweisungen zu ändern, was durch eine entsprechende Anpassung des Quellcodes von preProcess bewerkstelligt wurde.

Das Programm preProcess ist noch für einen weiteren wichtigen Schritt zuständig. Es handelt sich hierbei um das homogenisieren der Daten. Damit ist gemeint, dass, bevor Datensätze zum Trainieren weitergegeben werden, die Anzahl von Events innerhalb einer Klasse aneinander angepasst wird. Grund hierfür ist die Gefahr, dass ein Klassifikator, wenn er mit Daten trainiert wird, deren Klassenverteilung übermäßig unausgeglichen ist, unempfindlich für die Klasse mit weniger Events in den Trainingsdaten wird.

Zur Verdeutlichung ein einfaches Beispiel:
Es geht um die Klassifizierung UpDown und man hat 100000 Events zum Trainieren zur Verfügung. Innerhalb dieser Daten stammen 80% aus der Klasse „Down“ und 20% aus der Klasse „Up“. Wird nun ein RDF mit genau diesen Daten trainiert, so werden Events, bei denen eine große Unsicherheit bezüglich der Klassenzugehörigkeit besteht, mit hoher Wahrscheinlichkeit der Klasse „Down“ zugeordnet, da dies, wenn man von den Trainingsdaten ausgeht, die wahrscheinlichere Klasse des Events ist. Die Klassifizierung wird stark von der Verteilung der Events bei den Trainingsdaten beeinflusst. Es ist sinnvoller unsichere Ereignisse in eine extra Klasse für abgelehnte Events einzuordnen, um diese evtl. gesondert betrachten zu können.

4.3 Evaluationsmethoden

Da das Kernthema dieser Arbeit der Vergleich zwischen Klassifizierungssicherheiten eines Klassifikators mit unterschiedlichen Kennzahlenvektoren ist, spielt die Evaluation einer Klassifizierung eine entscheidende Rolle. Die Methoden, mit denen die Evaluierung für die berechneten Ergebnisse durchgeführt wurde, werden im Nachfolgenden beschrieben.

4.3.1 Kreuzvalidierung

Möchte man die Sicherheit, mit der ein Klassifikator Daten in Klassen unterteilt, berechnen, so ist es eminent wichtig, dass die Daten, die zu Evaluationszwecken verwendet werden, verschieden von den Daten sind, die zu Trainingszwecken verwendet wurden. Man benötigt also mindestens zwei Sätze von Daten, einen zum Trainieren und einen für die Evaluation. Dieses Resultat würde allerdings nur die Qualität des Klassifikators gemessen an einem Beispiel wiedergeben, was statistisch gesehen ein sehr unsicheres Ergebnis wäre. Um eine allgemeinere Aussage über einen Klassifikator machen zu können ist es deshalb notwendig, mit verschiedenen Daten zu Trainieren und aus allen einzelnen Resultaten eine Klassifikationssicherheit zu berechnen.

Die Methode, derer man sich bei dieser Arbeit bedient hat, heißt Kreuzvalidierung. Die Kreuzvalidierung basiert auf dem Prinzip, dass ein Teil der Daten zum Trainieren und ein anderer Teil für die Evaluation verwendet wird. Um nun eine möglichst allgemeine Aussage über die Effizienz eines Klassifikators machen zu können, werden vorhandene Daten wie folgt behandelt (vgl. Abbildung 4.1):

Zunächst werden alle Daten (N Events), die man für die Evaluation verwendet, per Zufall angeordnet. Dieser Schritt ist wichtig um systematische Effekte, wie z.B. Energie- und Winkelverteilung der Daten, zu umgehen. Nachdem die Daten zufällig verteilt sind, wird der Satz in n gleich große Anteile geteilt. Jeder Anteil besteht nun also aus N/n Events. Im nächsten Schritt wird der RDF mit den Events aus den ersten $n-1$ Anteilen trainiert und anschließend wird der Anteil, welcher nicht zum Trainieren verwendet wurde (also der $n$te Anteil) mit dem trainierten RDF klassifiziert. Das Resultat dieser Klassifizierung wird zwischengespeichert. Nun wird ein neuer RDF trainiert. Hierzu werden die Anteile 1 bis $n-2$ und n verwendet und im Anschluss wird der Teil $n-1$ klassifiziert und das Ergebnis wird ebenfalls gespeichert.

Dieses Schema wird nun sukzessiv weitergeführt, bis jeder der n Anteile der insgesamt N Events einmal klassifiziert worden ist. Abschließend werden alle zwischengespeicherten Ergebnisse zusammengerechnet. Mit Hilfe dieser Resultate lässt sich dann die Klassifizierungssicherheit folgendermaßen berechnen:

$$S = \frac{k}{N - R} \tag{4.1}$$

Hierbei steht k für die Summe aller korrekt klassifizierten Events, N für die Gesamtanzahl der Events und R für die Summe der vom Klassifikator abgelehnten Events.

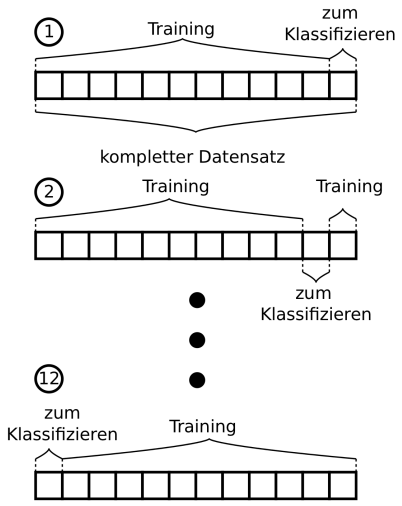

Abbildung 4.1: Schematische Darstellung der Kreuzvalidierung, wobei der komplette Datensatz in zwölf gleichgroße Anteile geteilt wurde

4.3.2 Klassifikationsübergreifende Auswertung

Mit der in diesem Kapitel beschriebenen Evaluationsmethode ist es möglich die Sicherheiten verschiedener Klassifizierungen miteinander zu vergleichen. Die Aussagen, die hierbei getroffen werden können, beziehen sich dann aber nur auf die jeweiligen Klassifikationen. Es fehlt also eine Methode um zu festzustellen, ob zusätzliche Kennzahlen aus einem bestimmten Modul die Sicherheiten von mehreren Klassifikationen im Durchschnitt erhöhen oder eher verringern. Im Rahmen dieser Arbeit wurde ein Ansatz verfolgt, welcher der Erweiterung des Kennzahlenvektors um Kennzahlen von einem bestimmten Modul einen Wert zuweist. Dieser Wert quantifiziert den Gewinn an Klassifikationssicherheit für Kennzahlen eines Moduls.

Intuitiv wäre es einen durchschnittliche Verbesserung der Klassifikationssicherheit für die Kennzahlen jedes Moduls zu berechnen. Hierzu vergleicht man die Sicherheiten, die mit zwei unterschiedlichen Kennzahlenvektoren bei der gleichen

Klassifikation berechnet wurden, miteinander. Bei den beiden Vektoren handelt es sich einmal um eine Version ohne die Kennzahlen eines bestimmten Moduls und einmal um den gleichen Vektor mit den zusätzlichen Kennzahlen. Pro Klassifikation wurden acht verschiedene Vektoren verwendet, was pro Modul zu vier Vektoren führt, bei denen die entsprechenden Kennzahlen verwendet wurden und vier ohne diese Kennzahlen. Damit sind pro Klassifizierungen vier Paare pro Modul auszuwerten. Bei der Energierekonstruktion müsste man also pro Klassifizierung die Sicherheiten der folgenden Vektoren jeweils miteinander vergleichen, bzw. die Differenz der Sicherheiten bilden: 000 und 100, 001 und 101, 010 und 110 sowie 011 und 111. Berechnet man nun pro Modul diese vier Werte für alle durchgeführten Klassifizierungen und bildet den Durchschnitt, so erhält man eine Gesamtaussage. Dieser Wert lässt aber eine wesentliche Eigenschaft der Klassifikationssicherheiten unberücksichtigt: der Wertebereich, in dem sich die Klassifikationssicherheit ohnehin schon befindet. Besitzt eine Klassifikation ein Sicherheit von $0,50$ so ist es wesentlich einfacher eine Steigerung dieser Sicherheit auf $0,55$ zu erreichen, als eine Steigerung von $0,90$ auf $0,95$ zu erzielen. Um diese Eigenschaft mit einzubeziehen wurde die durchschnittliche Sicherheit einer Klassifizierung in der Rechnung berücksichtigt.

Die Rechnung um klassifikationsübergreifende Aussagen zu erhalten wird nun am Beispiel der Energierekonstruktion dargestellt (andere Module analog).

Zunächst werden die Differenzen der Sicherheiten innerhalb einer Klassifikation berechnet:

$$S(100) - S(000) = \Delta_1 \tag{4.2}$$
$$S(101) - S(001) = \Delta_2 \tag{4.3}$$
$$S(110) - S(010) = \Delta_3 \tag{4.4}$$
$$S(111) - S(011) = \Delta_4 \tag{4.5}$$

Hierbei steht $S(100)$ für die Sicherheit die bei einer Klassifikation mit dem Vektor 100 erreicht wurde.

Im nächsten Schritt wird aus diesen vier Differenzen das arithmetische Mittel gebildet:

$$\overline{\Delta_k} = \frac{1}{4} \sum_{i=1}^{4} \Delta_i \tag{4.6}$$

Das k steht für die Klassifikation.

Bevor der eben berechnete Durchschnitt nun mit anderen Klassifikationen in Verbindung gebracht wird, wird ein Gewichtungsfaktor berechnet. Als Gewichtungsfaktor wird der reziproke durchschnittliche Fehler $\overline{F}^{-1}$ einer Klassifikation

berechnet:

$$\overline{S} = \frac{1}{8}[S(000) + S(001) + \cdots + S(111)] \tag{4.7}$$

$$\overline{F} = 1 - \overline{S} \tag{4.8}$$

Das Produkt aus reziprokem gemitteltem Fehler und durchschnittlicher Erhöhung der Sicherheit unter Verwendung der Kennzahlen aus einem Modul ergibt dann eine relative Verbesserung:

$$\Delta_{R,k} = \frac{\overline{\Delta_k}}{\overline{F_k}} \tag{4.9}$$

Für jede Klassifizierung lässt sich so eine relative Verbesserung pro zusätzlichem Kennzahlenblock berechnen. Um nun eine klassifikationsübergreifende Aussage über zusätzliche Kennzahlen machen zu können wird über alle relativen Verbesserungen gemittelt. Bei diesem Schritt wird berücksichtigt, wie oft welche Klassifikation auftritt. Bei dieser Arbeit wurden fünf verschiedene Energieklassifikationen verwendet (bezeichnet von $0E$ bis $4E$). Diese Klassifikationen wurden zwei bzw. drei Mal mit verschiedenen Teilchen trainiert verwendet. Zunächst wurden die relativen Verbesserungen für jede einzelne Klassifikation gemittelt:

$$\Delta_g = \frac{1}{J} \sum_{k=1}^{J} \Delta_{R,k} \tag{4.10}$$

Bei der Gleichung 4.10 ist J gleich 3 falls eine Klassifizierung einmal nur mit Myonen, einmal nur mit Neutrinos und einmal mit Myonen und Neutrinos durchgeführt wurde. J ist gleich zwei falls die Klassifizierung für Myonen ausblieb. Der Index g steht für den Klassifikationstyp, also z.B. Energieklassifikation $0E$ oder $1E$.

Abschließend wird über diese, für jeden Klassifikationstypen berechneten Werte gemittelt:

$$\Delta_G = \frac{1}{N} \sum_{g} \Delta_g \tag{4.11}$$

Hierbei steht N für die Anzahl der Klassifikationen und Δ_G wird als globale Verbesserung bezeichnet.

Innerhalb dieser Arbeit wurden fünf verschiedene Energieklassifikationen und eine UpDown Klassifikation untersucht. Für die Berechnung der globalen Verbesserung wurde allerdings nicht über alle sechs Klassifikationen gemittelt. Stattdessen wurden Energie- und UpDown Klassifikationen getrennt. Somit wurden für jedes Modul zwei globale Verbesserungen berechnet.

4.4 Selektion der wirkungsvollsten Kennzahlen

Wie in Abschnitt 3.1 bereits erwähnt ermöglicht ein größerer Kennzahlenvektor nicht zwingend bessere Klassifikationsergebnisse. Zusätzlich erhöht die Menge der Kennzahlen die Rechenzeit, die ein Klassifikator, sowohl beim Training als auch beim Klassifizieren benötigt. Diese beiden Punkte motivieren den folgenden Abschnitt, in welchem es um die Selektion der wirkungsvollsten Kennzahlen abhängig vom jeweiligen Klassifikationsproblem geht.

Diese Arbeit beschränkt sich auf das Berechnen eines Kennzahlenvektors mit den sechs wirkungsvollsten Kennzahlen. Bevor näher auf die Implementierung eingegangen wird, ist es sinnvoll zu erläutern was mit „wirkungsvoll“ gemeint ist. Eine wirkungsvolle Kennzahl ist eine Kennzahl, die möglichst viel Information über die Klassenzugehörigkeit der jeweiligen Daten beinhaltet und diese Information für den jeweiligen RDF auch zugänglich ist. Möchte man zum Beispiel zwischen Bananen und Kirschen unterscheiden, so ist die Farbe eine wesentlich sinnvollere Kennzahl als die elektrische Ladung.

Um nun zwischen „schlechten“ und „guten“ Kennzahlen unterscheiden zu können wurde das Programm „featureOptimize“ verwendet. Übergibt man featureOptimize einen Datensatz in Form von „.meta“ und „.dat“ Dateien, also einen Datensatz, dessen Kennzahlen bereits berechnet wurden und die Klassenzugehörigkeit in den „meta“ Dateien gespeichert ist, so wird zunächst die erste Kennzahl des Kennzahlenvektors verwendet. Mit nur dieser Kennzahl wird dann eine Kreuzvalidierung durchgeführt und die resultierende Klassifikationssicherheit wird festgehalten. Anschließend wird zur nächsten Kennzahl übergegangen und dieser Schritt mit nur dieser Kennzahl wiederholt. Ist man schließlich am Ende des Kennzahlenvektors angelangt, so wird die Kennzahl, welche die höchste Klassifikationssicherheit ergeben hat, festgehalten. Im Folgenden wird diese Kennzahl als $K1$ bezeichnet.

Im nächsten Schritt werden nun zwei Kennzahlen für die Kreuzvalidierung benutzt. Zum einen $K1$, welche beim ersten Durchlauf das beste Ergebnis erzielt hat, und zum anderen nacheinander jede andere Kennzahl im Kennzahlenvektor. Nachdem man den Kennzahlenvektor in dieser Weise durchlaufen hat, wird die Kennzahl, welche in Verbindung mit $K1$ die höchste Klassifikationssicherheit ergeben hat, ebenfalls festgehalten (und im Folgenden als $K2$ bezeichnet). Dieses Schema wird so lange wiederholt, bis die vorher definierte Anzahl an Kennzahlen (hier sechs) gefunden wurde.

Diese Methode liefert aber nicht unbedingt die Untermenge der Kennzahlen, die tatsächlich zum besten Klassifikationsergebnis führen würden, sondern es handelt sich nur um eine lokale Optimierung. Es ist durchaus möglich, dass eine andere Kombination von Kennzahlen ein besseres Ergebnis liefern würde. Das Problem hierbei ist aber große Anzahl an möglichen Kombinationen. Bei einem Kennzahlenvektor von 189 Kennzahlen und der Suche nach einer Untermenge mit sechs

Kennzahlen sind dass $5,84 \cdot 10^{10}$ Möglichkeiten, wobei die Klassifikationssicherheit jeder dieser Varianten mittels Kreuzvalidierung ermittelt werden müsste, was den Rahmen des machbaren deutlich übersteigt.

4.5 Arbeitsfluss

In dem folgenden Abschnitt soll noch einmal kurz der gesamte Arbeitsfluss, welcher in den letzten Kapiteln in einzelne Schritte zerlegt beschrieben wurde, übersichtlich dargestellt werden. In Abbildung 4.2 sind die einzelnen Arbeitsschritte schematisch abgebildete.

Der erste Schritt beinhaltet die Übergabe der Simulationen an I3RDFFeature-Extract. Wie bereits beschrieben wurde I3RDFFeatureExtract so angepasst, dass zusätzliche Module aufgerufen werden, welche die neuen Kennzahlen berechnen. Die berechneten Kennzahlen werden anschließend in „X.dat“ und „X.meta“ gespeichert („X“ steht hier stellvertretend für eine frei wählbaren Dateinamen).

Als nächstes werden „X.dat“ und „X.meta“ an preProcess übergeben. preProcess berechnet nun für jedes Event die entsprechende Klassenzugehörigkeit. Diese Information wird in „X.meta“ gespeichert. Zusätzlich werden die Kennzahlenvektoren entsprechend zurechtgeschnitten und in „X.dat“ geschrieben. Pro Klassifizierung und Kennzahlenvektor wird eine „X.dat“ und eine „X.meta“ erzeugt, was für sechs Klassifizierungen und acht verschiedene Kennzahlenvektoren 48 Kombinationen ergibt.

Die nächsten beiden Schritte werden unabhängig von einander durchgeführt. Eine Aufgabe besteht darin bei allen 48 Kombinationen die Klassifizierungssicherheit mittels Kreuzvalidierung (durchgeführt von dem Programm „crossValidation“) zu ermitteln.

Die andere Aufgabe ist es, für jede Klassifizierung einen optimierten Kennzahlenvektor zu finden. Für diese Aufgabe wird pro Klassifizierung nur eine (nämlich die des größten Kennzahlenvektors) „X.dat“ und eine „X.meta“ an featureOptimize übergeben, da man die besten sechs Kennzahlen aus allen zur Verfügung stehenden Kennzahlen ermitteln möchte.

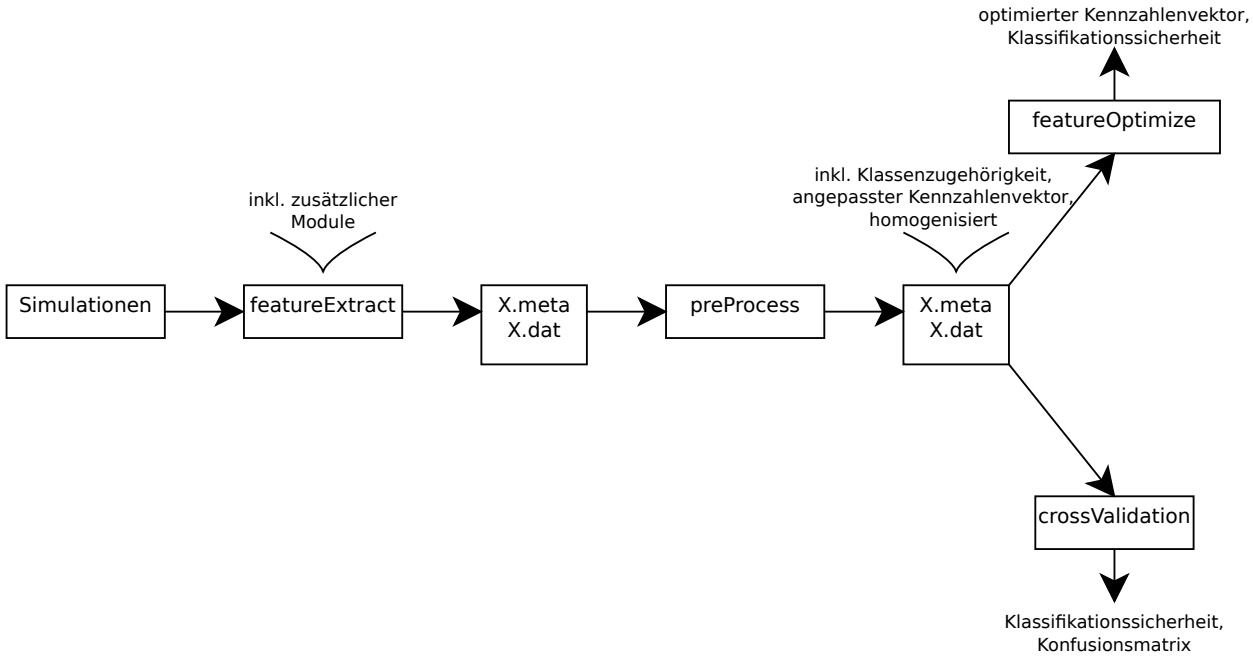

Abbildung 4.2: Darstellung der Arbeitsschritte

Kapitel 5

Ergebnisse

Das Kernthema dieser Arbeit ist die Erweiterung des Kennzahlenvektors, welcher von dem Klassifikator I3RDFClassify benutzt wird und die Untersuchung, ob sich Klassifikationssicherheiten, unter Benutzung des erweiterten Kennzahlenvektors, bei verschiedenen Klassifikationen verändern. Wie in Abschnitt 4.4 beschrieben wurde für jede Klassifikation aus allen verwendeten Kennzahlen eine Untermenge von sechs möglichst optimalen Kennzahlen berechnet.

In dem folgenden Kapitel sind die Ergebnisse der Kreuzvalidierungen aller Untersuchten Klassifikationen dargestellt. Außerdem werden die optimierten Kennzahlenvektoren für die einzelnen Klassifikationen angegeben.

5.1 Darstellung der Ergebnisse

Wie in Kapitel 4 beschrieben, stammen die zusätzlich verwendeten Kennzahlen aus drei verschiedenen Modulen. Da die Beeinflussung der Klassifikationssicherheiten möglichst differenziert untersucht werden sollte, wurden acht verschiedene Kennzahlenvektoren pro Klassifikation verwendet. Hierbei handelt es sich um den Standardkennzahlenvektor, so wie er vor dieser Arbeit benutzt wurde, und um sieben erweiterte Vektoren (siehe Abschnitt 4.2.3). Die Kennzahlenvektoren werden mit dreistelligen Binärcodes referenziert, wobei jede Stelle für die zusätzlichen Kennzahlen aus einem Modul steht. Die Reihenfolge ist dabei: AntEnergyReco - DusjShowerReco - rrFitReco. Steht an der entsprechenden Stelle eine 1 so wurden die Kennzahlen verwendet, bei einer 0 wurden die Kennzahlen nicht verwendet. Um das Spektrum der Evaluation noch weiter aufzufächern wurde jede Klassifikation drei Mal durchgeführt. Einmal wurden nur Myonen zum Trainieren und Klassifizieren verwendet, einmal nur Neutrinos und anschließend sowohl Myonen, also auch Neutrinos. Für jede Klassifikation und jeden Kennzahlenvektor wurden also drei Klassifikationssicherheiten berechnet. Motivation hierfür war das Ausgleichen

eventueller systematischer Unterschiede in den Simulationen. Jede der berechneten Sicherheiten in diesem Abschnitt stammt aus einer Kreuzvalidierung mit zehn Durchläufen.

Die Klassifikationssicherheiten für die einzelnen Klassifikationsprobleme sind in jeweils einem Graphen in Form eines Histogrammes abgebildet. Ein Graph besteht also immer aus acht Balken, je ein Balken für einen Kennzahlenvektor. Um die Übersichtlichkeit der Graphiken zu steigern wurden verschiedene Farben verwendet, abhängig davon welche Teilchen verwendet wurden: Myonen (blau), Neutrinos (grün), Myonen und Neutrinos (rot).

5.2 Klassifikation nach Energieklassen

5.2.1 Definition der Energieklassen

Bei fünf der sechs untersuchten Klassifikationen handelt es sich um Einteilungen in Energieklassen. Hierbei handelt es sich um zwei bis vier Klassen.

Die Bezeichnungen der Energieklassifikationen und die Definition der Klassen lautet wie folgt:

- Bezeichnung: 0E - Grenzen($200\,GeV|3\,TeV$)
  - Klasse 0: $E < 200\,GeV$
  - Klasse 1: $200\,GeV < E < 3\,TeV$
  - Klasse 2: $3\,TeV < E$
- Bezeichnung: 1E - Grenzen($100\,GeV|1\,TeV|5\,TeV$)
  - Klasse 0: $E < 100\,GeV$
  - Klasse 1: $100\,GeV < E < 1\,TeV$
  - Klasse 2: $1\,TeV < E < 5\,TeV$
  - Klasse 3: $5\,TeV < E$
- Bezeichnung: 2E - Grenze($100\,TeV$)
  - Klasse 0: $E < 1\,TeV$
  - Klasse 1: $1\,Tev < E$
- Bezeichnung: 3E - Grenze($1\,PeV$)
  - Klasse 0: $E < 1\,PeV$
  - Klasse 1: $1\,PeV < E$

- Bezeichnung: 4E - Grenze($10\,PeV$)
  - Klasse 0: $E < 10\,PeV$
  - Klasse 1: $10\,PeV < E$

5.2.2 Klassifikationssicherheiten bei Einteilung in Energieklassen

5.2.2.1 Klassifikation 0E ($200\,GeV|3\,TeV$)

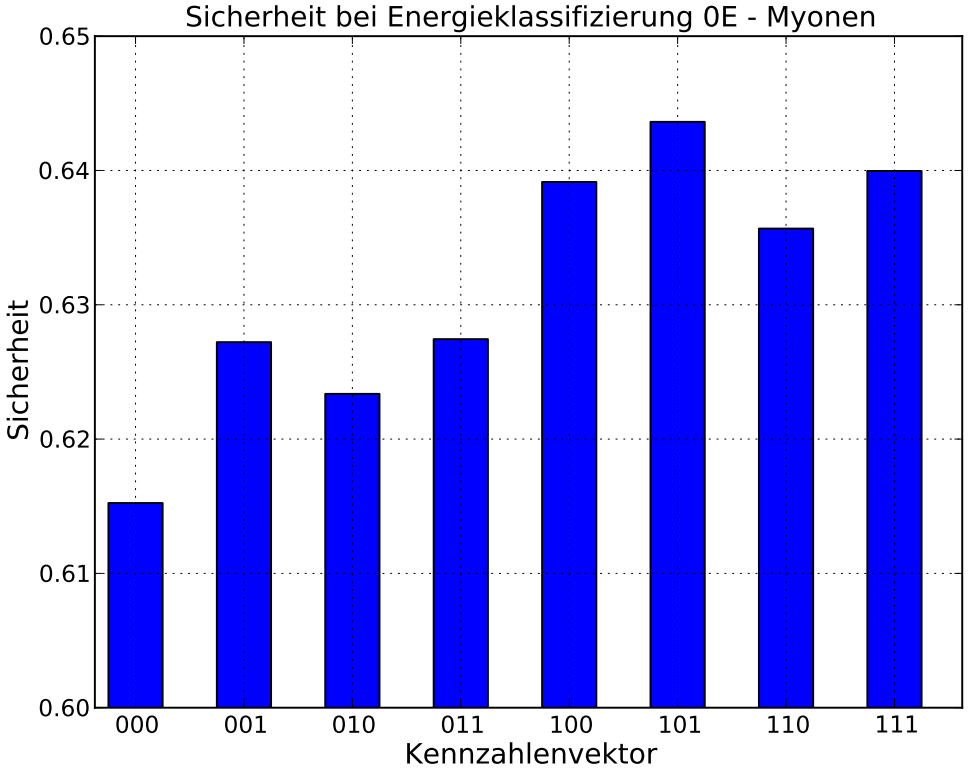

Abbildung 5.1: Klassifikationssicherheit für Myonen bei Klassifizierung 0E

In Abbildung 5.1 sind die Klassifikationssicherheiten für die Klassifizierung 0E dargestellt, wobei hier nur Myonen verwendet wurden. Es wurde mit durchschnittlich 9400 Events evaluiert. Die höchste Klassifikationssicherheit lag bei $S_{max} =$

$0,643619$ und wurde mit dem Kennzahlenvektor 101 erzielt. Die niedrigste Klassifikationssicherheit ergab die Klassifizierung mit dem Vektor 000 und beträgt $S_{min} = 0,615238$. Hieraus ergibt sich eine maximale Differenz von $\Delta_S = 0.028381$.

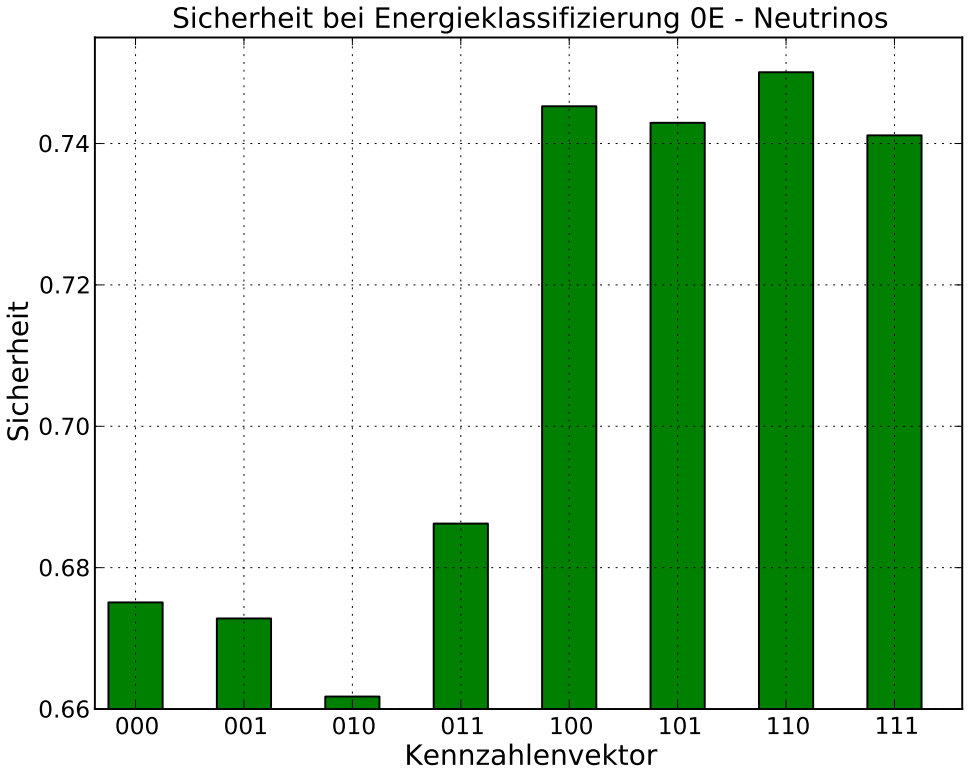

Abbildung 5.2: Klassifikationssicherheit für Neutrinos bei Klassifizierung 0E

In Abbildung 5.2 sind die Klassifikationssicherheiten von 0E unter Verwendung von Neutrinos dargestellt. Bei der Evaluation wurde mit ca. 3010 Events gearbeitet. Mit dem Kennzahlenvektor 110 wurde die größte Sicherheit von $S_{max} = 0,750086$ erreicht und die niedrigste Sicherheit von $S_{min} = 0,661765$ wurde mit dem Vektor 010 erzielt. Die maximale Differenz beträgt damit $\Delta_S = 0,088321$.

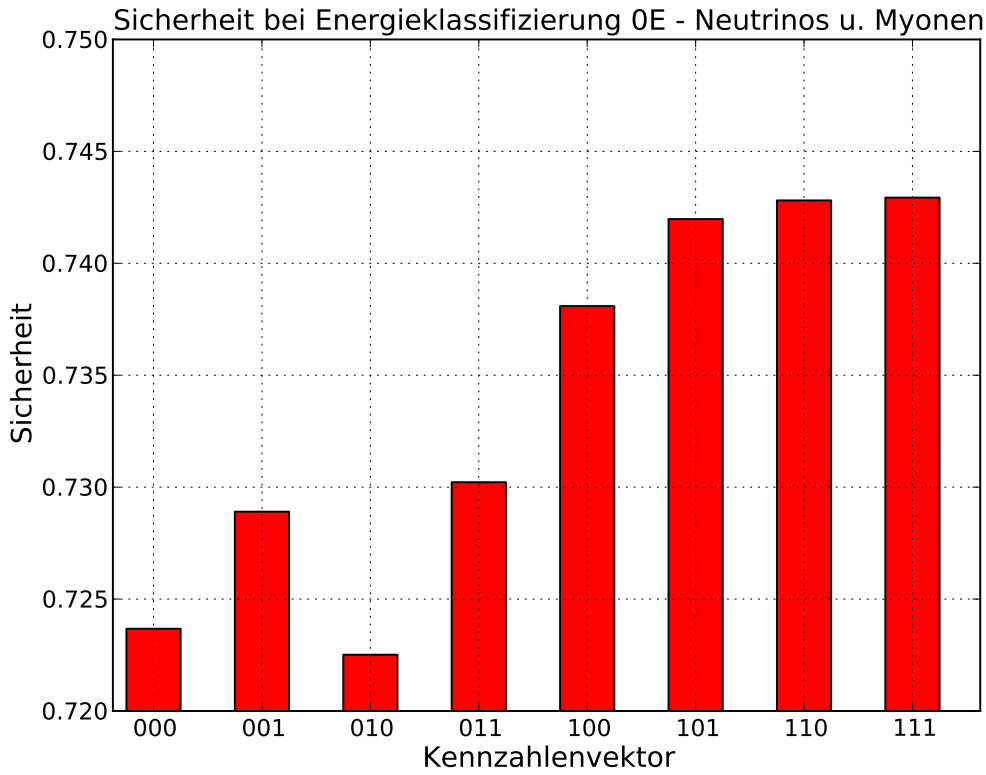

Abbildung 5.3: Klassifikationssicherheit für Neutrinos und Myonen bei Klassifizierung 0E

Die Klassifikationssicherheiten unter Verwendung von Neutrinos und Myonen ist in Abbildung 5.3 abgebildet. Hierbei wurde im Durchschnitt mit ca. 18840 Events evaluiert. Unter Verwendung des Kennzahlenvektors 111 wurde die höchste Sicherheit von $S_{max} = 0,742935$ erreicht. Die Verwendung des Vektors 010 führte zur geringsten Sicherheit von $S_{min} = 0,722514$, womit sich eine maximale Differenz von $\Delta_S = 0,020421$ berechnen lässt.

5.2.2.2 Klassifikation 1E ($100\,GeV|1\,TeV|5\,TeV$)

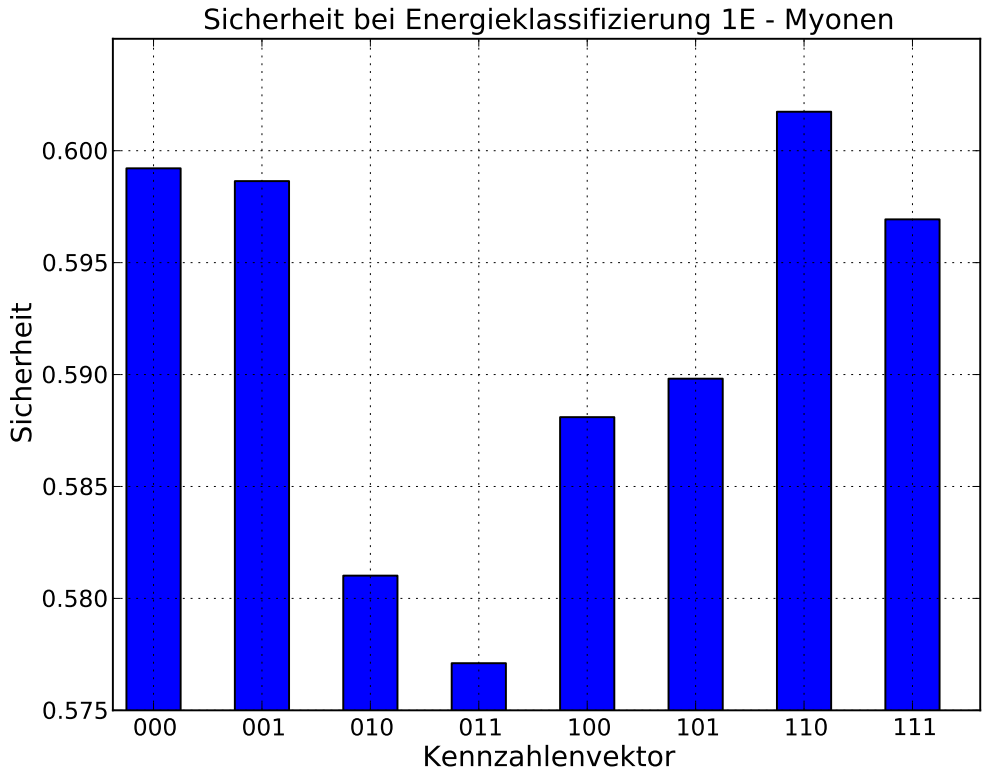

Abbildung 5.4: Klassifikationssicherheit für Myonen bei Klassifizierung 1E

Die Abbildung 5.4 stellt die Klassifikationssicherheiten von 1E unter Verwendung von Myonen dar. Für diese Klassifizierung wurden im Schnitt 2540 Events verwendet. Die höchste Klassifikationssicherheit von $S_{max} = 0,601741$ wurde unter Verwendung des Vektors 110 erreicht, die niedrigste Sicherheit mit $S_{min} = 0,577105$ ergab sich unter Verwendung von 011. Die maximale Differenz ergibt somit $\Delta_S = 0,024636$.

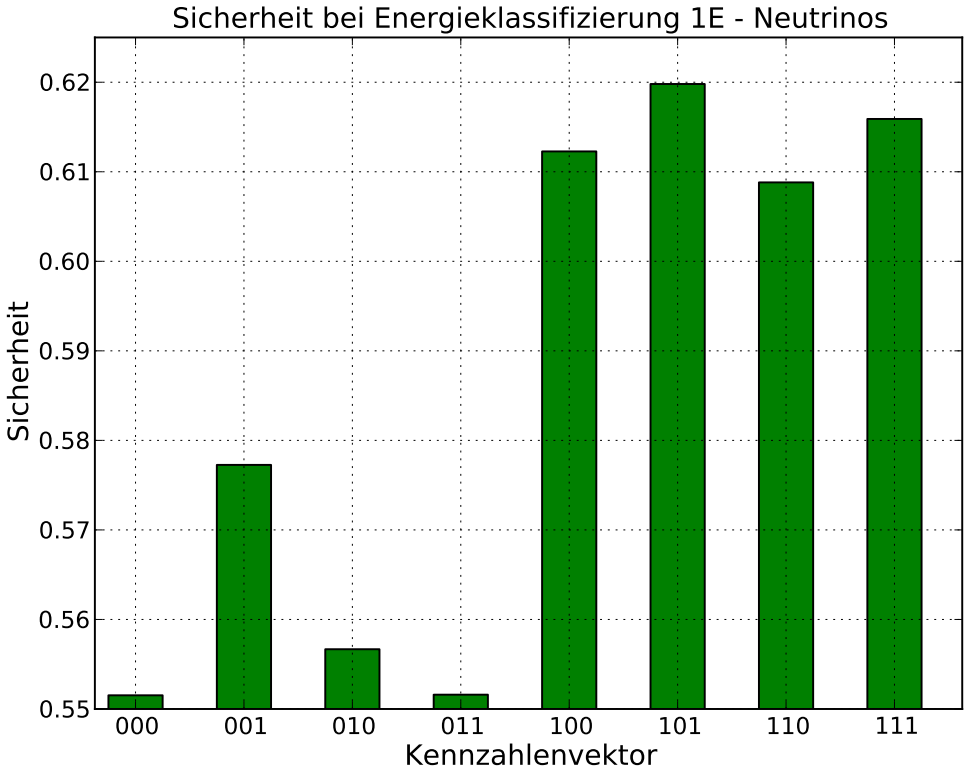

Abbildung 5.5: Klassifikationssicherheit für Neutrinos bei Klassifizierung 1E

Die Klassifizierung 1E unter Verwendung von Neutrinos ist in Abbildung 5.5 abgebildet. Bei dieser Klassifizierung wurden ca. 2290 Events verwendet. Mit dem Kennzahlenvektor 101 wurde die höchste Sicherheit von $S_{max} = 0,619801$ erzielt, mit dem Vektor 000 die geringste mit einem Wert von $S_{min} = 0,551528$. Hieraus lässt sich ein maximaler Unterschied von $\Delta_S = 0,068273$ berechnen.

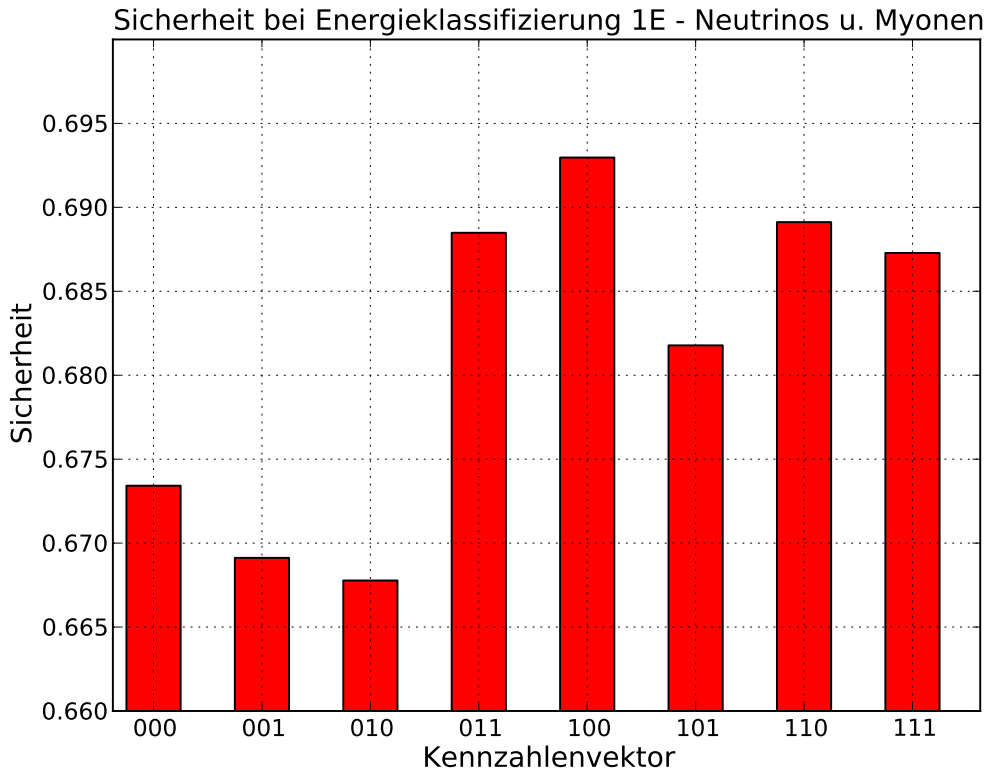

Abbildung 5.6: Klassifikationssicherheit für Neutrinos und Myonen bei Klassifizierung 1E

Unter Verwendung von Neutrinos und Myonen ergab die Klassifizierung 1E die in Abbildung 5.6 dargestellten Sicherheiten. Hierbei wurden jeweils ca. 4840 Events verwendet. Der Kennzahlenvektor 100 erbrachte die höchste Sicherheit mit $S_{max} = 0,692966$. Die niedrigste Klassifikationssicherheit wurde mit dem Vektor 010 erreicht und beträgt $S_{min} = 0,667772$, womit sich eine maximale Differenz von $\Delta_S = 0,025194$ ergibt.

5.2.2.3 Klassifikation 2E ($100\,TeV$)

Ab der Energieklassifikation 2E konnten keine Sicherheiten nur unter Verwendung von ausschließlich Myonen berechnet werden, da diese einen Energiebereich von über $100\,TeV$ nur sehr selten erreichen (siehe Abbildung D.2) und sich mit einer zu geringen Anzahl an Events kein Klassifikator mehr sinnvoll trainieren lässt.

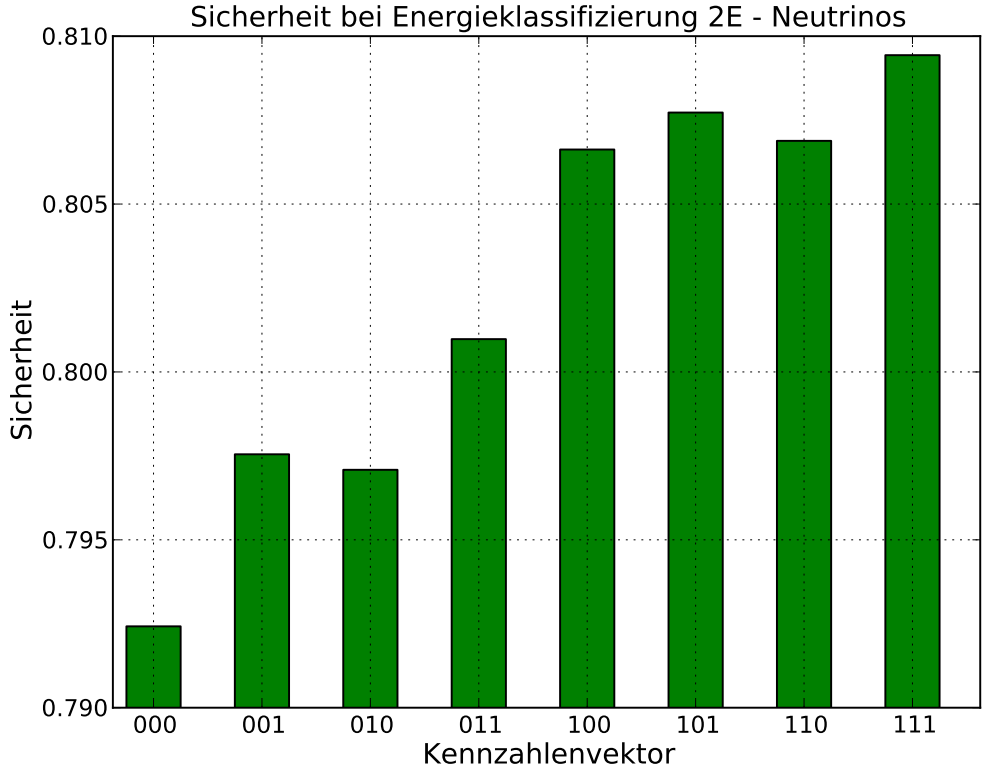

Abbildung 5.7: Klassifikationssicherheit für Neutrinos bei Klassifizierung 2E

In Abbildung 5.7 sind die Klassifikationssicherheiten für die Klassifizierung 2E unter Verwendung von Neutrinos abgebildet, wobei im Schnitt ca. 27000 Events verwendet wurden. Die höchste Sicherheit von $S_{max} = 0,809434$ wurde mit dem Vektor 111 erreicht. Die niedrigste Sicherheit beträgt $S_{min} = 0,792423$ und wurde bei der Verwendung des Vektors 000 berechnet. Der maximale Unterschied der Sicherheiten liegt somit bei $\Delta_S = 0,017011$.

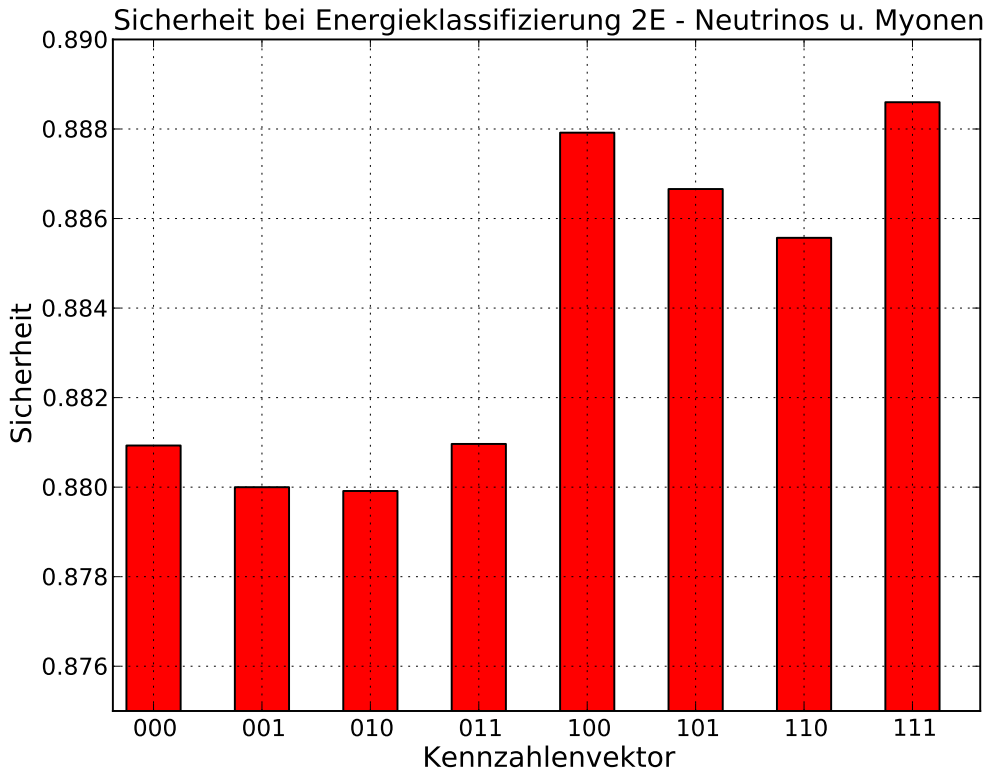

Abbildung 5.8: Klassifikationssicherheit für Neutrinos und Myonen bei Klassifizierung 2E

Bei Verwendung von Neutrinos und Myonen ergeben sich die in Abbildung 5.8 dargestellten Sicherheiten für 2E. Bei diesen Klassifizierungen wurden jeweils ca. 59400 Events verwendet. Bei Verwendung des Vektors 111 ergab sich $S_{max} = 0,888598$. Die niedrigste Sicherheit $S_{min} = 0,879914$ wurde bei Verwendung des Vektors 010 berechnet. Hiermit ergibt sich eine maximale Differenz von $\Delta_S = 0,008684$.

5.2.2.4 Klassifikation 3E ($1\,PeV$)

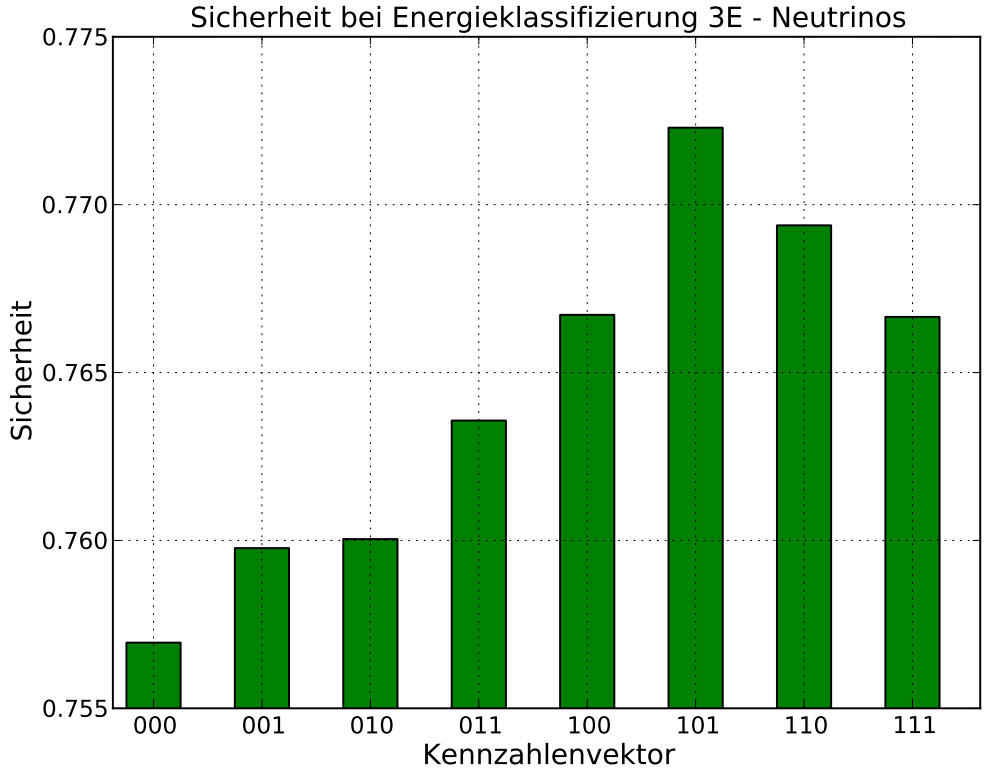

Abbildung 5.9: Klassifikationssicherheit für Neutrinos bei Klassifizierung 3E

In Abbildung 5.9 sind die Sicherheiten für 3E bei der Verwendung von Neutrinos abgebildet. Hierbei wurden je ca. 37100 Events zur Evaluation verwendet. Die maximale Sicherheit von $S_{max} = 0,772293$ wurde unter Verwendung von 101 erzielt und die minimale Sicherheit von $S_{min} = 0,756956$ mit 000. Damit ergibt sich eine maximale Differenz der Sicherheiten von $\Delta_S = 0,015337$

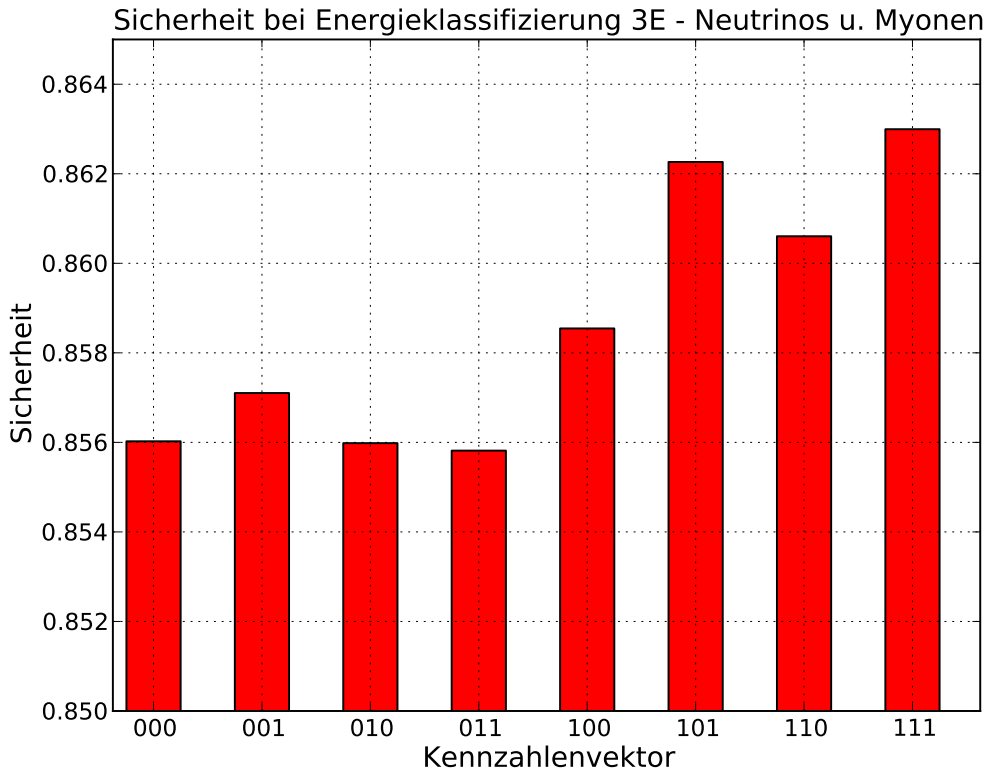

Abbildung 5.10: Klassifikationssicherheit für Neutrinos und Myonen bei Klassifizierung 3E

Die Klassifikationssicherheiten für Neutrinos und Myonen bei der Klassifizierung 3E ist in Abbildung 5.10 dargestellt. Es wurden für diese Klassifizierung ca. 49400 Events verwendet. Die höchste Sicherheit von $S_{max} = 0,862995$ wurde unter Verwendung des Vektors 111 erreicht und die niedrigste Sicherheit von $S_{min} = 0,855816$ wurde unter Benutzung des Vektors 011 berechnet. Hieraus ergibt sich ein maximaler Unterschied von $\Delta_S = 0,007179$.

5.2.2.5 Klassifikation 4E ($10\,PeV$)

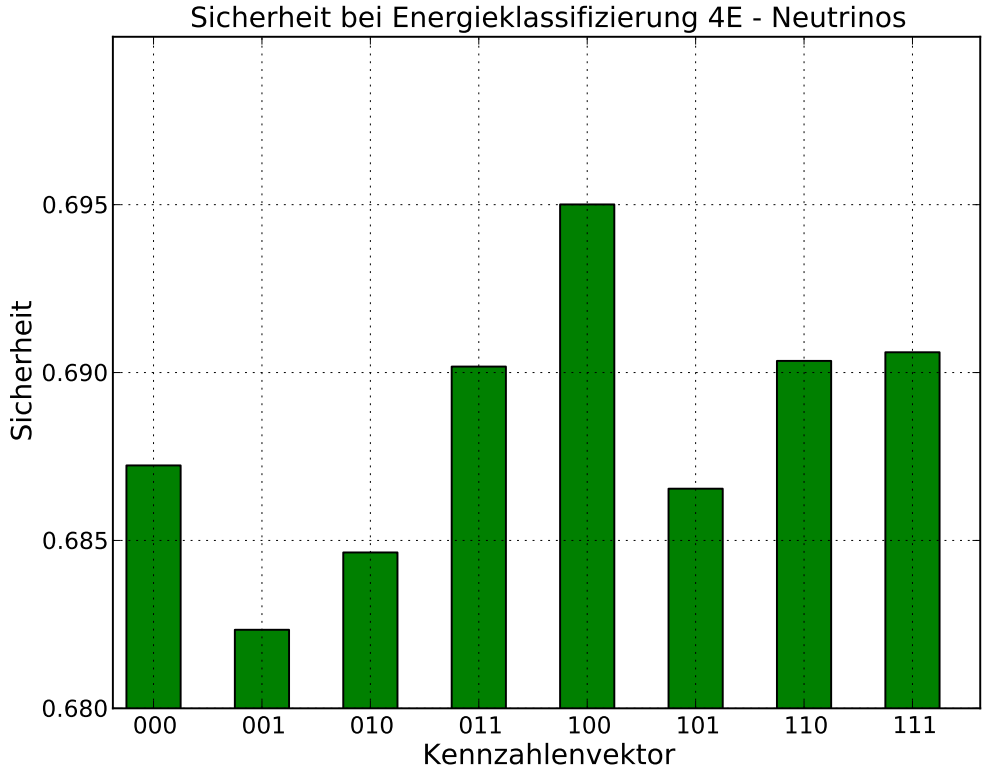

Abbildung 5.11: Klassifikationssicherheit für Neutrinos bei Klassifizierung 4E

Abbildung 5.9 zeigt die berechneten Sicherheiten für die Klassifizierung 4E unter Verwendung von Neutrinos. Es wurden jeweils ca. 15900 Events für die Berechnungen verwendet. Unter Benutzung des Kennzahlenvektors 100 wurde $S_{max} = 0.695007$ berechnet und $S_{min} = 0.682339$ ergab sich bei Verwendung des Vektors 001. Somit ist die maximale Differenz $\Delta_S = 0,012668$.

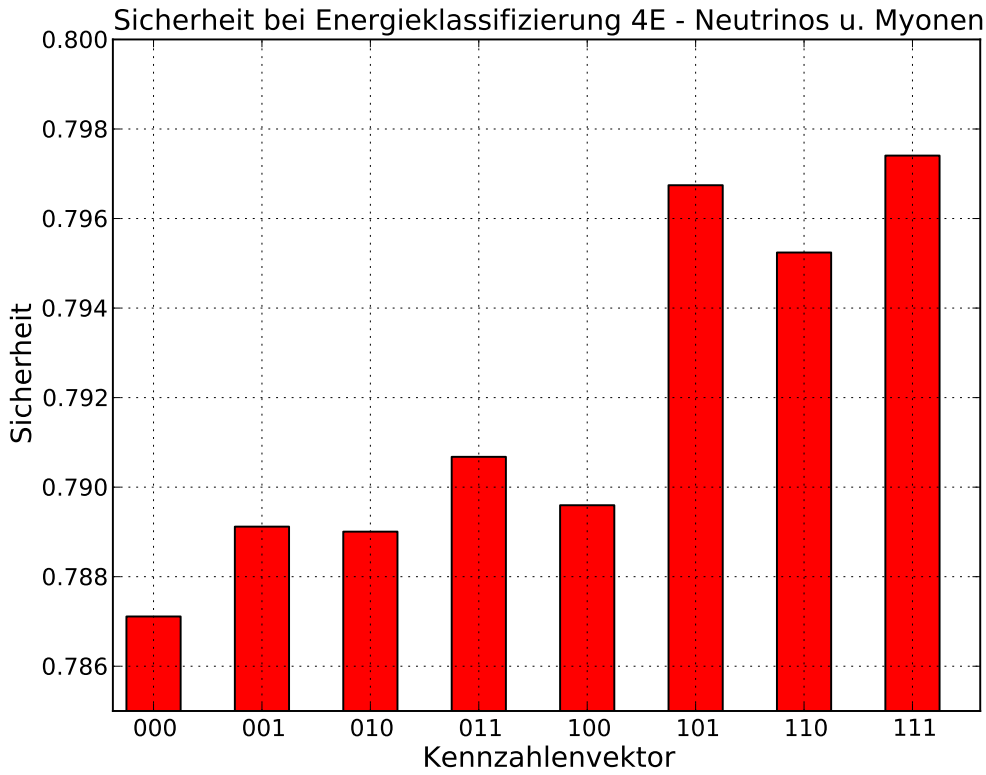

Abbildung 5.12: Klassifikationssicherheit für Neutrinos und Myonen bei Klassifizierung 4E

Die in Abbildung 5.12 dargestellten Sicherheiten ergaben sich bei Verwendung von Neutrinos und Myonen für die Klassifizierung 4E. Für diese Berechnungen wurden jeweils ca. 15900 verwendet. Die maximale Sicherheit von $S_{max} = 0,797406$ bei Verwendung des Vektors 111 während die Verwendung des Vektors 000 zur minimalen Sicherheit $S_{min} = 0,787111$ führte. Die damit berechnete maximale Differenz beträgt $\Delta_S = 0,010295$.

5.3 UpDown Klassifizierung

Bei der Berechnung der Klassifizierungssicherheiten für die UpDwon Klassifizierung wurden sowohl Neutrinos als auch Myonen verwendet. Eine Aufspaltung bei dieser Klassifikation ist nur wenig sinnvoll, da alle simulierten Neutrinos nur von unten nach oben fliegen und alle simulierten Myonen nur von oben kommen.

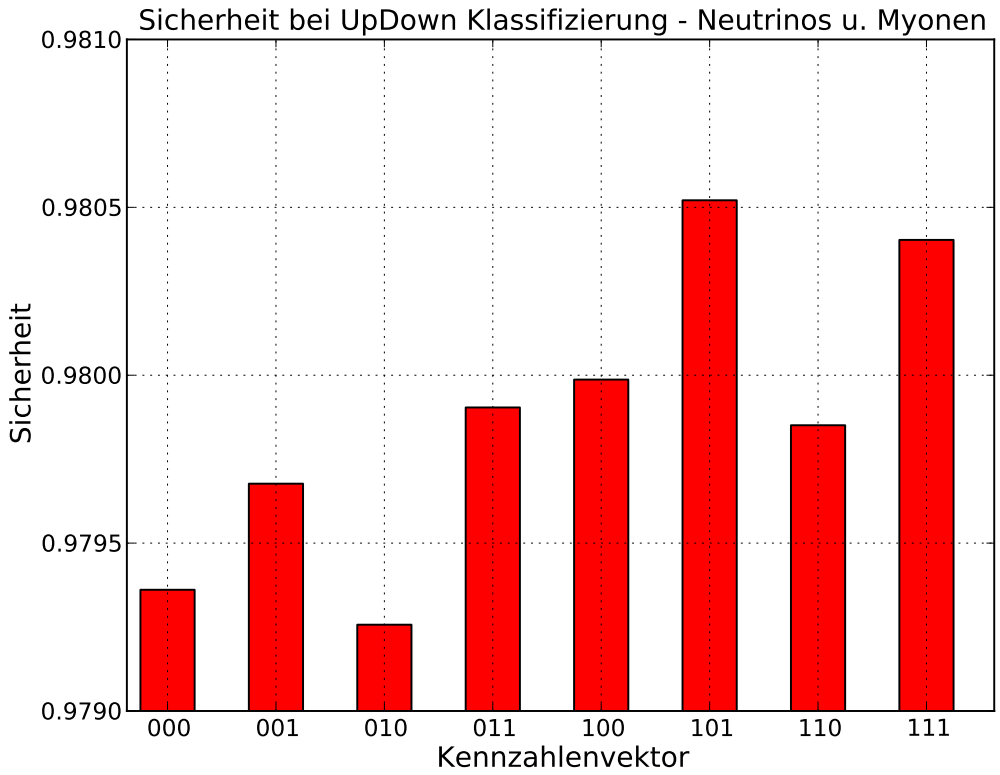

Abbildung 5.13: Klassifikationssicherheit für Neutrinos und Myonen bei UpDown Klassifizierung

Abbildung 5.13 stellt die Ergebnisse für die UpDown Klassifizierung unter Verwendung von Neutrinos und Myonen dar. Bei dieser Klassifizierung wurden jeweils ca. 82000 Events verwendet. Die höchste Klassifikationssicherheit von $S_{max} = 0,980403$ wurde mit Verwendung des Vektors 111 erzielt. Die niedrigste Sicherheit $S_{min} = 0,979257$ wurde mit dem Vektor 010 berechnet. Damit ergibt sich eine maximale Differenz von $\Delta_S = 0,001146$.

5.4 Globale Verbesserungen

Die in 4.3.2 beschriebenen globalen Verbesserungen ergaben für die Energieklassifikationen folgende Ergebnisse:

	AntEnergyReco	DusjShowerReco	rrFitReco
Δ_G	0.057980	0.002899	0.009281

Tabelle 5.1: Globale Verbesserung für Energieklassifikationen

Für die UpDown Klassifikation ließen sich folgende Werte berechnen:

	AntEnergyReco	DusjShowerReco	rrFitReco
Δ_G	0.031831	-0.001627	0.025447

Tabelle 5.2: Globale Verbesserung für die UpDown Klassifikation

5.5 Optimierte Kennzahlenvektoren

Um die wichtigsten Kennzahlen zu identifizieren wurde unter allen zur Verfügung stehenden Kennzahlen eine Untermenge von sechs Kennzahlen ermittelt, mit welchen sich ein lokales Maximum für die Sicherheit erzielen lässt. Diese Optimierung wurde für alle zwölf im letzten Abschnitt vorgestellten Klassifikationsprobleme durchgeführt. Die Ergebnisse sind in Tabelle 5.3 zusammengefasst (Tabelle der optimierten Vektoren C.1).

Klassifikation	Teilchen	Events	S_{max}	S_{opt}	Δ
0E	Myonen	9500	$0,643619$	$0,571564$	$0,072055$
1E	Myonen	2600	$0,601741$	$0,594967$	$0,006774$
0E	Neutrinos	3000	$0,750086$	$0,741856$	$0,00823$
1E	Neutrinos	2300	0.619801	0.638988	$-0,019187$
2E	Neutrinos	27000	$0,809434$	$0,800252$	$0,009182$
3E	Neutrinos	37100	$0,772293$	$0,760949$	$0,011344$
4E	Neutrinos	15900	$0,695007$	$0,687834$	$0,007173$
0E	Myonen u. Neutrinos	18800	$0,742935$	$0,720391$	$0,022544$
1E	Myonen u. Neutrinos	4800	$0,692966$	$0,670056$	$0,02291$
2E	Myonen u. Neutrinos	29800	$0,888598$	$0,876657$	$0,011941$
3E	Myonen u. Neutrinos	24700	$0,862995$	$0,851715$	$0,01128$
4E	Myonen u. Neutrinos	16000	$0,797406$	$0,748778$	$0,048628$
UD	Myonen u. Neutrinos	41400	$0,980403$	$0,976111$	$0,004292$

Tabelle 5.3: Sicherheiten der optimierten Kennzahlenvektoren

Die Spalte „Events“ gibt an wie viele Ereignisse jeweils für die Klassifikation verwendet wurden (gerundet). S_{max} steht für die maximale Sicherheit, welche für die jeweilige Klassifikation unter Verwendung von allen Kennzahlen erzielt werden konnte. Dieser Wert ist also gleichzustellen mit S_{max} des letzten Abschnittes.

In der Spalte S_{opt} ist die maximale Sicherheit dargestellt, welche unter Verwendung der optimierten Kennzahlenvektoren mit lediglich sechs Kennzahlen erreicht wurde. Abgesehen von der Klassifikation 0E für Myonen und 4E für Neutrinos und Myonen wurde S_{opt} immer mit dem optimierten Kennzahlenvektor mit sechs Kennzahlen berechnet, da dieser zur höchsten Sicherheit führte. Bei 0E mit Myonen wurde das lokale Optimum der Sicherheit mit einem Kennzahlenvektor, der nur drei Kennzahlen enthielt erreicht. Bei 4E für Neutrinos und Myonen erzielte der Vektor mit nur fünf Kennzahlen das lokale Optimum.

Zusätzlich wurde die Differenz zwischen S_{max} und S_{opt} in der Spalte Δ angegeben ($\Delta = S_{max} - S_{opt}$).

Kapitel 6

Diskussion

In dem letzten Kapitel wurden die Ergebnisse präsentiert, die aufzeigen ob zusätzliche Kennzahlen eine höhere Klassifikationssicherheit ermöglichen. Im folgenden Kapitel sollen diese Ergebnisse erklärt oder plausibel gemacht werden. Zusätzlich werden die Daten hinsichtlich der Frage, ob das Hinzufügen von Kennzahlen die erreichbaren Klassifikationssicherheiten erhöht, interpretiert.

Bevor die Ergebnisse näher erläutert werden, soll kurz auf die statistische Sicherheit der berechneten Werte eingegangen werden. An mehreren Stellen innerhalb der verwendeten Algorithmen werden von Zufallsgeneratoren erzeugte Zahlen verwendet. Eine dieser Stellen befindet sich innerhalb des Programms zur Kreuzvalidierung. Hierbei wird auf Zufallszahlen zurückgegriffen um zu entscheiden welche Events zum Trainieren und welche zum Klassifizieren verwendet werden. Außerdem wird innerhalb der RDFs von Zufallszahlen Gebrauch gemacht, unter anderem um zu entscheiden welche Kennzahlen bei den einzelnen Entscheidungsbäumen verwendet werden. Die Tatsache, dass Zufallszahlen verwendet werden, sorgt für eine gewisse Schwankung, was die Sicherheiten von Klassifikationen angeht.

Um eine grobe Vorstellung von diesen Schwankungen zu bekommen wurde jede Kreuzvalidierung ein zweites Mal ausgeführt, wobei pro Durchlauf einer Kreuzvalidierung ohnehin je zehn Klassifikationen mit unterschiedlichen Zufallszahlen durchgeführt werden. Bei dem zweiten Durchlauf wurde ein anderer Seed für den verwendeten Zufallsgenerator verwendet, was dafür sorgt, dass andere Zufallszahlen verwendet werden. Die Ergebnisse dieser Referenzrechnung wurden mit den im letzten Kapitel dargestellten verglichen und sind ebenfalls in Histogrammform in Anhang A zu finden. Der Vergleich zwischen den dargestellten Ergebnissen und den Referenzwerten zeigte Schwankungen von $0,01\%$ bis hin zu Schwankungen von $< 1,1\%$. Hierbei konnte ebenfalls festgestellt werden, dass diese Schwankungen abhängig von der durchschnittlichen Sicherheit der Klassifizierung sind. Je höher die Sicherheit, desto geringer die Schwankungen. Des weiteren traten Schwankungen von über 1% nur bei den Klassifizierungen $0E$ trainiert mit Neutrinos, $1E$ trainiert

mit Neutrinos und $1E$ trainiert mit Myonen auf. Alle diese Klassifizierungen haben gemein, dass nur sehr wenige Events zum Trainieren verwendet wurden. Trotz dieser Schwankungen konnten, bis auf wenige Ausnahmen, die gleichen Tendenzen für die Klassifikationssicherheiten unter Verwendung der einzelnen Kennzahlenvektoren auch bei den Referenzdaten beobachtet werden.

6.1 Energieklassifikationen

Zunächst werden die Ergebnisse diskutiert, bei denen an den Standardkennzahlenvektor jeweils alle Kennzahlen aus den Modulen AntEnergyReco, DusjShowerReco und rrFitReco angehängt wurden. Bei fünf der sechs untersuchten Klassifikationen handelt es sich um Einteilungen in Energieklassen.

Die globale Verbesserung für zusätzlichen Kennzahlen aus der Energierekonstruktion weist für Energieklassifikationen einen Wert von $0,057980$ auf und ist damit um eine Größenordnung höher, als die der Kennzahlen der anderen Module. Bereits beim ersten Blick auf die erstellten Graphen fällt auf, dass die letzten vier Balken in fast allen Fällen höher sind, als die ersten vier. Des weiteren sind innerhalb der Kennzahlenvektoren, die bei den Klassifikationen die höchsten Sicherheiten erzielen, immer die Kennzahlen der Energierekonstruktion vertreten. Zusätzlich sind die Kennzahlenvektoren, welche die schlechtesten Ergebnisse produzieren, immer ohne die zusätzlichen Kennzahlen aus der Energierekonstruktion. Die These, dass Kennzahlen aus einer Energierekonstruktion positive Effekte bei einer Energieklassifikation erzeugen, kann durch die berechneten Daten untermauert werden.

Für die Kennzahlen der Module DusjShowerReco und rrFitReco ergeben sich für die globalen Verbesserungen ebenfalls positive Werte, diese sind aber deutlich geringer als die von AntEnergyReco. Dies kann, unter Betrachtung der Graphen, leicht nachvollzogen werden.

Die Ursache für den mehr als dreimal so hohen Wert der globalen Verbesserung von rrFitReco gegenüber DusjShowerReco kann folgendermaßen erklärt werden. Zunächst muss das Energiespektrum der klassifizierten Teilchen berücksichtigt werden. Die simulierten Myonen sind im Durchschnitt wesentlich niederenergetischer als Neutrinos. Bei einer Untersuchung der Spektren hat sich ergeben, dass die Energien von simulierten Myonen eine obere Grenze bei einer Energie von ca. $1\,PeV$ besitzen. Simulierte Neutrinos hingegen erreichen Energien von über $10\,PeV$. Die Energiespektren der Teilchen sind dem Anhang beigefügt (Anhang D). Zusätzlich muss die Eigenschaft der Simulationen berücksichtigt werden, dass atmosphärische Myonen nur von oben kommen und Neutrinos nur von unten. Kombiniert mit den Energieeigenschaften der Teilchen ergibt sich somit ein Zusammenhang zwischen Teilchenenergie und Teilchenspur. Da sich rrFitReco mit Spurrekonstruktion be-

schäftigt, kann der berechnete Wert für die globale Verbesserung bei Energieklassifikationen gegenüber dem Wert für DusjShowerReco auf diese Weise plausibel gemacht werden. In Abbildung 5.12 tritt dieser Effekt sehr anschaulich auf. Die Klassifizierung $4E$ trennt in zwei Klassen mit einer Grenze bei $10\,PeV$. Wie eben erwähnt gibt es keine simulierten Myonen oberhalb von $1\,PeV$, weshalb gerade bei diesem Beispiel nicht nur nach Energie sondern auch nach Teilchen und damit nach Teilchenspur getrennt wird. Vergleicht man jeden zweiten Balken mit seinem linken Nachbarn, so stellt man fest, dass die Balken, welche Kennzahlen aus rrFitReco benutzen, in allen Fällen höher sind. Dieses Verhalten lässt sich auch in den Referenzdaten erkennen.

Bei der Auswertung der Energieklassifikationen sind auch einige überraschende Ergebnisse aufgetreten. Ein Beispiel hierfür ist die Klassifikation $1E$ unter Verwendung von Neutrinos. Betrachtet man die Sicherheiten der Vektoren 001, 010 und 011, so stellt man fest, dass die Erweiterung des Standardvektors um Kennzahlen aus rrFitReco einen Erhöhung der Sicherheit erzielt. Die Erweiterung des Standardvektors mit Kennzahlen aus DusjShowerReco erzielt ebenfalls einen Gewinn an Klassifikationssicherheit, wenn auch einen geringeren. Intuitiv erwartet man nun, dass das Hinzufügen der Kennzahlen aus beiden Modulen ebenfalls eine Erhöhung der Klassifikationssicherheit bewirkt, jedoch ist dies nicht der Fall. Stattdessen sinkt die Sicherheit wieder nahezu auf den Wert des Standardvektors zurück. Eine stichhaltige Begründung für dieses Verhalten kann nicht dargelegt werden. Diese Ergebnisse sind allerdings vor dem Hintergrund zu sehen, dass bei der Klassifizierung lediglich 2290 Events verwendet wurden. Für wirklich aussagekräftige Ergebnisse wären mindestens 10000 Events nötig gewesen. Bei den für diese Arbeit berechneten Simulationen waren nur sehr wenige Neutrinos im Energiebereich von unter $100\,GeV$ vorhanden. Die Berechnung von mehr Simulationen konnte wegen des hohen Rechenaufwands, der dafür nötig wäre, nicht mehr durchgeführt werden. Des weiteren handelt es sich bei gerade dieser Klassifikation um eine der drei Klassifikationen, bei denen sich im Vergleich mit den Referenzdaten eine Abweichung von knapp über einem Prozent bei einer Sicherheit ergeben hat. In sofern könnte man das unerwartete Verhalten mit statistischen Schwankungen begründen. Dem entgegen spricht allerdings das Auftreten des gleichen Phänomens bei den Referenzdaten (siehe Abbildung A.5).

Abschließend lässt sich sagen, dass eine Erweiterung des Kennzahlenvektors um Kennzahlen aus einer Energierekonstruktion nachweislich bessere Klassifikationsergebnisse ermöglicht. Dies konnte in allen getesteten Energieklassifikationen gezeigt werden. Eine Erweiterung um die Kennzahl aus dem Modul rrFitReco hat global gesehen zwar auch einen Gewinn an Sicherheit ermöglicht, ist aber erst bei Klassifikationen mit hochenergetischen Grenzen wirklich ausschlaggebend. Die Erweiterung des Kennzahlenvektors mit den Kennzahlen aus dem Modul DusjS-

howerReco lieferte ebenfalls eine positive globale Verbesserung, welche allerdings deutlich geringer ist als die der beiden anderen Module. Der Informationsgewinn für eine Energieklassifikation aus den Kennzahlen dieses Moduls hat bei keiner der untersuchten Klassifikationen eine ausschlaggebende Erhöhung der Klassifikationssicherheit ergeben. Unter Berücksichtigung der enormen Rechenzeit, die benötigt wird, um die Kennzahlen aus DusjShowerReco zu berechnen, ist die Erweiterung um diese Kennzahlen für Energieklassifikationen eher ungeeignet.

6.2 UpDown Klassifikation

Die Klassifikation nach von oben und von unten kommenden Teilchen wurde getrennt von den Energieklassifikationen betrachtet. Außerdem wurde diese Klassifikation nur unter Verwendung von Neutrinos und Myonen evaluiert.

Auch bei dieser Klassifikation fällt auf, dass die besten Ergebnisse die Kennzahlen aus der Energierekonstruktion enthalten, was sich deutlich im Wert der globalen Verbesserung von $0,031831$ widerspiegelt. Dies legt die Schlussfolgerung nahe, dass eine Erweiterung des Kennzahlenvektors mit Kennzahlen aus AntEnergyReco die Klassifikationssicherheit für UpDown Klassifikationen erhöht. Grund hierfür ist mit hoher Wahrscheinlichkeit der in Abschnitt 6.1 erläuterte Zusammenhang zwischen Energie und Richtung, bzw. Teilchen.

Der ebenfalls sehr hohe Wert der globalen Verbesserung von $0,025447$ für die Kennzahl aus rrFitReco ist auch auffällig. Dies lässt sich auch innerhalb des Histogramms dieser Klassifikation erkennen (Abbildung 5.13). Vergleicht man wieder jeden zweiten Balken mit seinem linken Nachbarn, so stellt man fest, dass dieser immer kleiner ist. Dies unterstützt den intuitiven Gedanken, dass die Kennzahl, die auf einer Spurrekonstruktion basiert, bei einer UpDown Klassifikation verwendbare Informationen liefert.

Die Kennzahlen des Moduls DusjShowerReco ermöglichen bei dieser Klassifikation keinen Gewinn der Klassifikationssicherheit, was sich im negativen Wert der globalen Verbesserung für diese Kennzahlen widerspiegelt. Betrachtet man allerdings den Graphen der Referenzdaten dieser Klassifikation so kann man sehen, dass die Sicherheiten der Vektoren die DusjShowerReco verwenden etwas höher sind als die ohne diese Kennzahlen. Anhand der vorhandenen Ergebnisse lässt sich nicht eindeutig bestimmten ob diese Kennzahlen einen positiven Effekt für die UpDown Klassifizierung haben. Es kann aber davon ausgegangen werden, dass eine Beeinflussung der Klassifikationssicherheit, unabhängig ob positiv oder negativ, nur eine geringe Rolle spielt. Dies lässt sich am geringen betragsmäßigen Wert der globalen Verbesserung festmachen. In Anbetracht der hohen Rechenzeit für diese Kennzahlen ist deren Einbeziehung bei einer Klassifikation diesen Typs wohl nicht rentabel.

6.3 Optimierte Kennzahlenvektoren

Neben der Untersuchung ob eine Erweiterung des Kennzahlenvektors bessere Ergebnisse liefert, wurde unter den maximal zur Verfügung stehenden Kennzahlen auch ein optimierter Kennzahlenvektor aus nur sechs Kennzahlen berechnet.

Die in Tabelle 5.3 zusammengefassten Ergebnisse zeigen, dass sich die mit dem wesentlich kleineren optimierten Kennzahlenvektor erzielten Sicherheiten von bis zu $7,2\%$ von den Sicherheiten unterscheiden, die mit den erweiterten Vektoren berechnet wurden. Bei einer so großen Diskrepanz zwischen den erreichbaren Sicherheiten stellt der optimierte Kennzahlenvektor keine wirkliche Alternative dar. Die Ergebnisse dieser Untersuchung lassen aber Rückschlüsse auf den Informationsgehalt verschiedener Kennzahlen zu, was hilfreich bei der Berechnung eines optimalen Kennzahlenvektors sein könnte.

Bei der UpDown Klassifizierung unter Verwendung von Myonen und Neutrinos lies sich mit dem optimierten Kennzahlenvektor eine Sicherheit von $97,6\%$ erzielen, was nur um $0,4\%$ geringer ist als die maximal erreichbare Sicherheit unter Verwendung aller Kennzahlen. Bei dieser Klassifizierung wären weitere Suchen nach optimierten Kennzahlenvektoren mit der selben Größe durchaus interessant, da sich evtl. eine Kombination von Kennzahlen findet lässt, welche noch näher an dem mit allen Kennzahlen erzieltem Ergebnis ist.

Ein besonderes Interesse erweckt das Ergebnis der Klassifikation $1E$, mit Neutrinos trainiert, bei welcher der optimierte Kennzahlenvektor bereits ein höheres Ergebnis erzielte als das beste unter Verwendung von allen Kennzahlen. Dieses Resultat ist allerdings in Anbetracht der geringen Anzahl an Events mit der trainiert wurde mit Vorsicht zu betrachten.

Kapitel 7

Zusammenfassung und Ausblick

Im Laufe dieser Arbeit wurde untersucht, ob das Hinzufügen von zusätzlichen Kennzahlen zu einer bestehenden Menge an Kennzahlen die Sicherheit, die bestimmte Klassifikationen erzielen, erhöht. Hierbei handelt es sich um fünf Klassifikationen in Energieklassen und eine Klassifikation, die nach von oben kommende und von unten kommende Teilchen unterscheidet. Als Klassifikator wurden Random Decision Forests benutzt, welche in dem bestehenden Programm I3RDFClassify, eingebettet im Seatray Framework, implementiert sind. Die zusätzlichen Kennzahlen stammen ebenfalls aus drei Modulen des Seatray Frameworks, namentlich AntEnergyReco, DusjShowerReco und rrFitReco.

Es hat sich gezeigt, dass die Kennzahlen der Energierekonstruktion AntEnergyReco bei beiden Klassifikationstypen den größten Zuwachs der Klassifikationssicherheit ermöglichen.

Die zusätzliche Kennzahl aus der Spurrekonstruktion rrFitReco erzielte bei der getesteten UpDown Klassifizierung ebenfalls einen Gewinn was die Klassifikationssicherheit angeht. Bei Energieklassifikationen ist die Kennzahl aus diesem Modul erst bei hochenergetischen Energiegrenzen zwischen den Klassen sinnvoll. Dies hängt mit der Tatsache zusammen, dass innerhalb der Simulationen nur Neutrinos so hohe Energien erreichen und diese bei den verwendeten Simulationen immer von unten kommen.

Zusätzliche Kennzahlen aus der Schauerrekonstruktion ermöglichten weder bei Energieklassifikationen noch bei der UpDown Klassifikation einen bedeutenden Zuwachs an Sicherheit.

Neben dem Effekt, den eine Erweiterung des Kennzahlenvektors erzeugt, wurde für jede Klassifizierung die Sicherheit unter Benutzung eines verkürzten Kennzahlenvektors berechnet. Hierbei wurde sich auf einen Vektor von sechs Kennzahlen mit möglichst viel Informationsgehalt beschränkt. Abgesehen von einer Klassifizierung konnten mit diesen Vektoren Sicherheiten erreicht werden, die bei über 90% der Sicherheiten, welche unter Verwendung aller Kennzahlen berechnet wurden,

lagen.

In zukünftigen Versionen von I3RDFClassify sind Kennzahlen der Module AntEnergyReco und rrFitReco mit großer Wahrscheinlichkeit vertreten, da diese einen deutlichen Gewinn an Klassifikationssicherheit ermöglichen. In Anbetracht dieser Ergebnisse ist der Gedanke nicht weit, dass Kennzahlen aus weiteren, noch nicht in Betracht gezogenen Modulen ebenfalls zu besseren Resultaten bei der Klassifizierung von Events führen könnten. Eine Untersuchung dieser These erscheint viel versprechend.

Einen hohen Stellenwert bei der Weiterentwicklung der Ereignisklassifizierung nimmt wohl der Bereich zur Optimierung des Kennzahlenvektors ein. Bereits die hohen Sicherheiten, die mit einem Bruchteil der vorhandene Kennzahlen in dieser Arbeit berechnet wurden, deuten darauf hin, dass es möglich ist sehr gut Ergebnisse mit einem deutlich kleineren, optimierten Kennzahlenvektor zu erzielen. Bei einer Klassifikation innerhalb dieser Arbeit wurde unter Verwendung des verkürzten Kennzahlenvektors ein besseres Ergebnis erzielt, als unter Verwendung aller Kennzahlen. Dies deutet darauf hin, dass die Suche nach einem optimalen Kennzahlenvektor nicht nur mit der Motivation Rechenzeit einzusparen betrieben wird, sondern auch um die Klassifikationssicherheit an sich zu verbessern.

Bei manchen Klassifikationen wurden Ergebnisse berechnet, welche so nicht erwartet wurden. Hierbei handelt es sich um die scheinbare Beeinflussung der Kennzahlen aus verschiedenen Modulen untereinander (siehe z.B. Abbildung 5.5). Da bei diesen Klassifikationen aber nur sehr wenige Events verwendet wurden, kann man statistische Effekte nicht ausschließen. Eine genauere Untersuchung dieser Klassifikationen mit mehr Daten könnte auch dieses Phänomen erklären.

Danksagungen

Ich möchte mich an dieser Stelle kurz bei all jenen bedanken, die bei der Entstehung dieser Arbeit beteiligt waren.
Zunächst geht mein Dank an Frau Professor Anton für die Vergabe des Themas.
Außerdem möchte ich mich bei Dr. Thomas Eberl bedanken, der für Fragen und Ratschläge immer ein offenes Ohr hatte.
Besonders möchte ich mich auch bei Kathrin Roensch, Florian Folger und Roland Richter bedanken, die keine Mühen gescheut haben mir bei Fragen und Problemen zu helfen.
Zu guter Letzt geht ein großes Dankeschön an meinen Betreuer Stefan Geißelsöder, welcher mir zu jeder Tages- und Nachtzeit tatkräftig zur Seite stand.
Vielen Dank euch allen!

Anhang A

Graphen der Referenzdaten

Im folgenden sind die Histogramme der Klassifikationssicherheiten der zweiten Durchführung der Kreuzvalidierungen abgebildet.

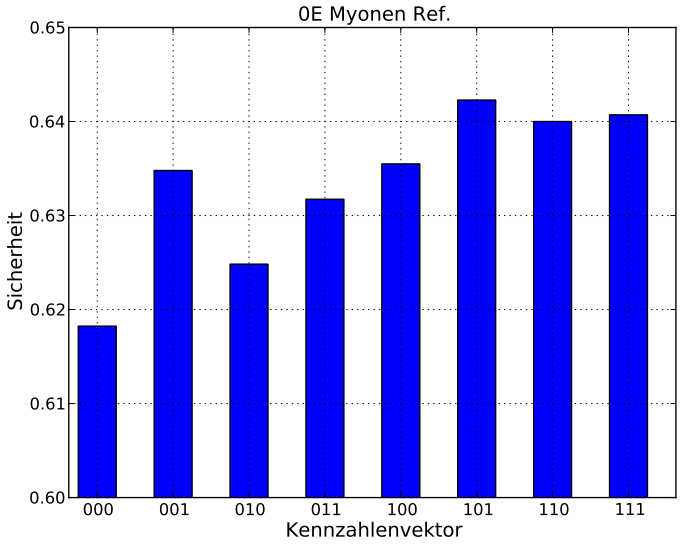

Abbildung A.1: Klassifikationssicherheit bei Klassifizierung 0E für Myonen

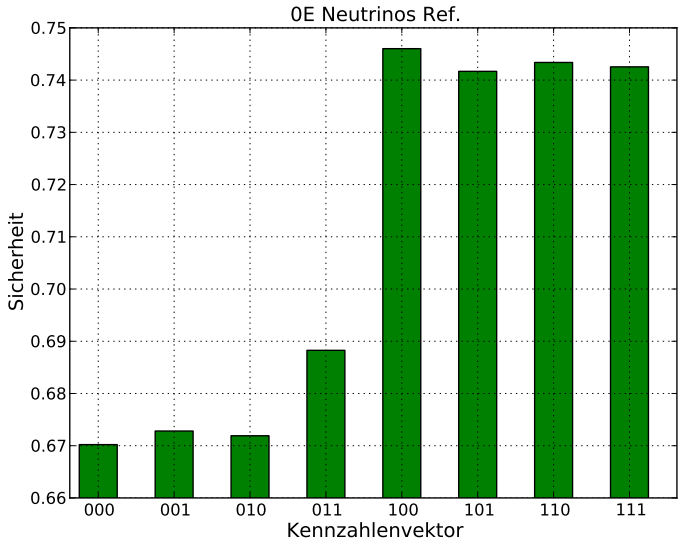

Abbildung A.2: Klassifikationssicherheit bei Klassifizierung 0E für Neutrinos

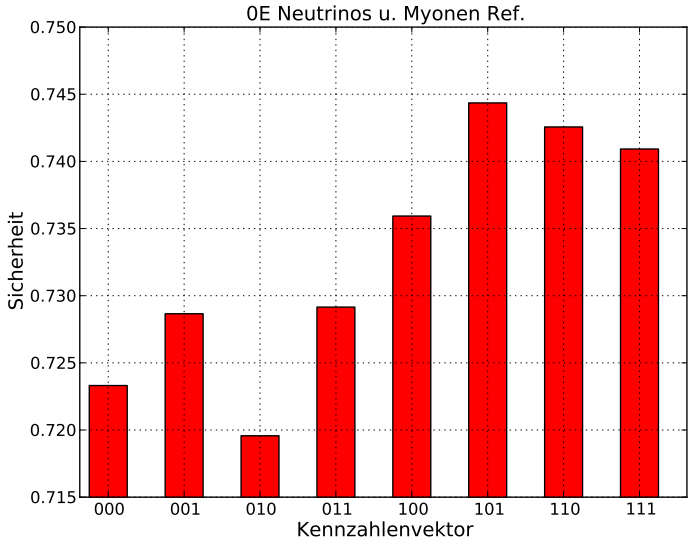

Abbildung A.3: Klassifikationssicherheit bei Klassifizierung 0E für Myonen und Neutrinos

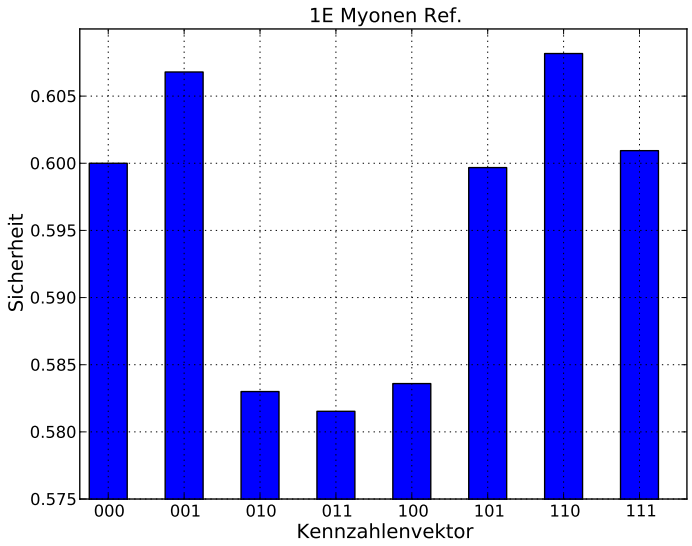

Abbildung A.4: Klassifikationssicherheit bei Klassifizierung 1E für Myonen

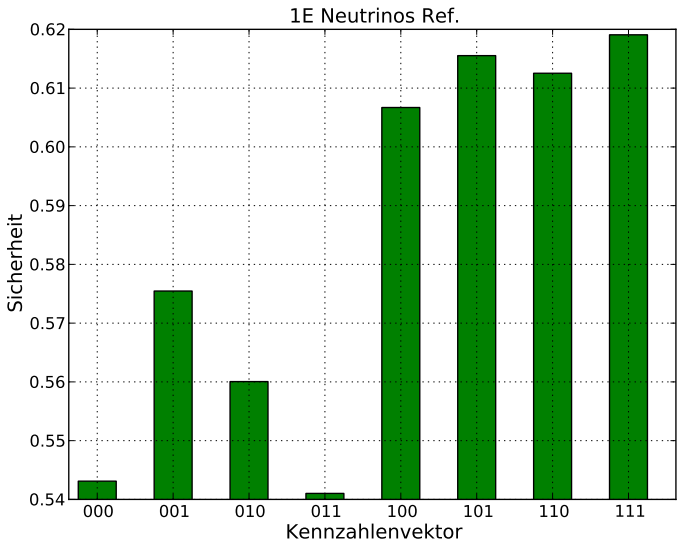

Abbildung A.5: Klassifikationssicherheit bei Klassifizierung 1E für Neutrinos

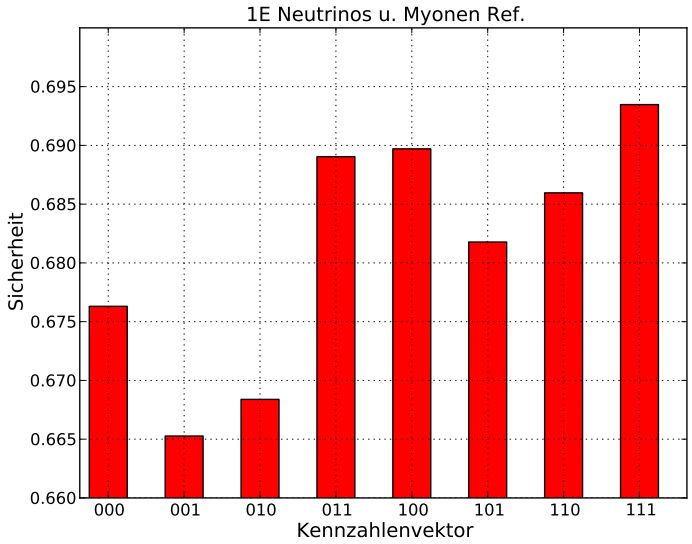

Abbildung A.6: Klassifikationssicherheit bei Klassifizierung 1E Myonen und Neutrinos

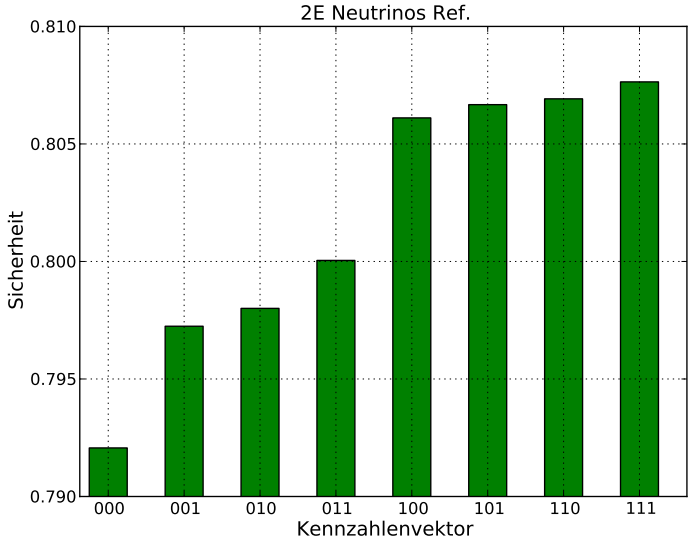

Abbildung A.7: Klassifikationssicherheit bei Klassifizierung 2E für Neutrinos

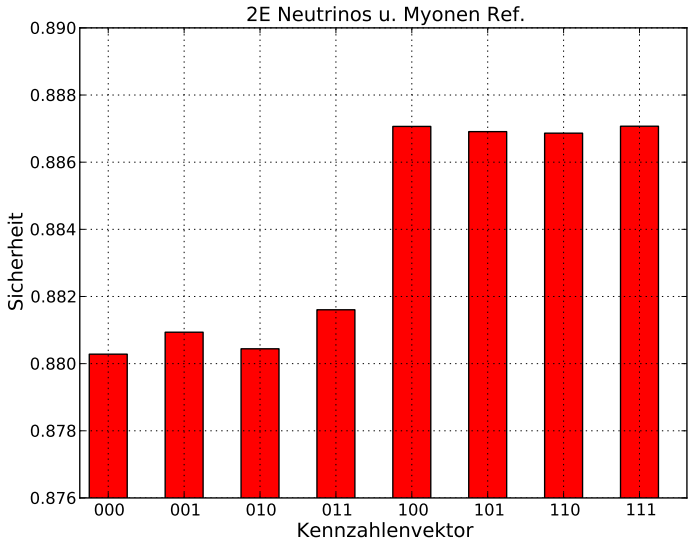

Abbildung A.8: Klassifikationssicherheit bei Klassifizierung 2E für Myonen und Neutrinos

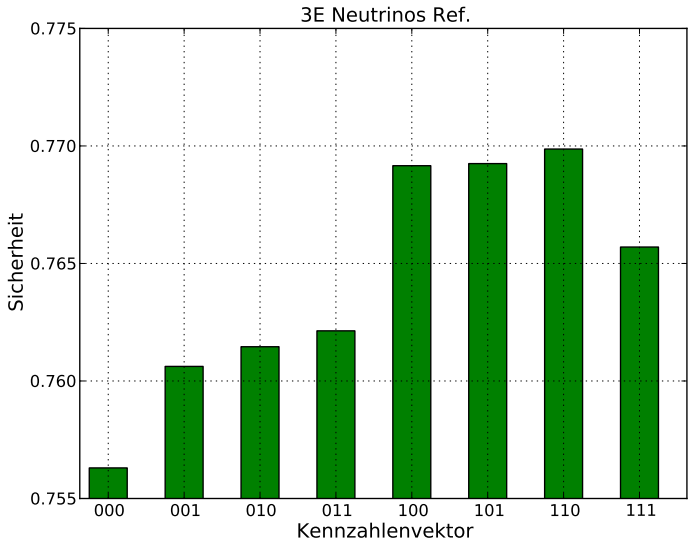

Abbildung A.9: Klassifikationssicherheit bei Klassifizierung 3E für Neutrinos

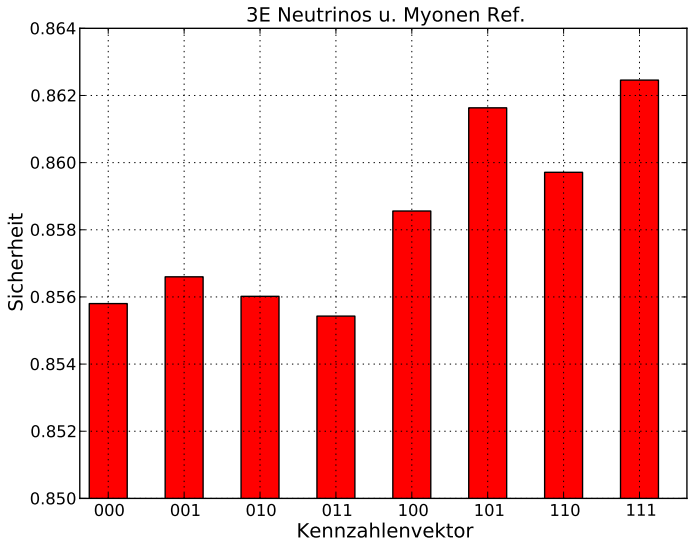

Abbildung A.10: Klassifikationssicherheit bei Klassifizierung 3E für Neutrinos und Myonen

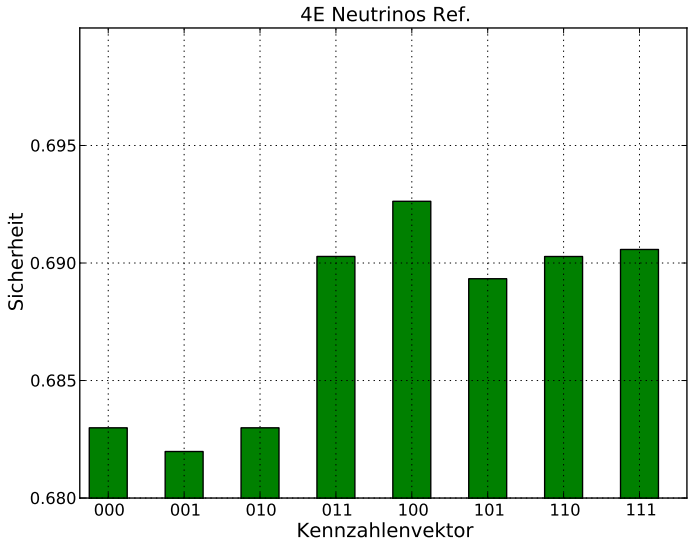

Abbildung A.11: Klassifikationssicherheit bei Klassifizierung 4E für Neutrinos

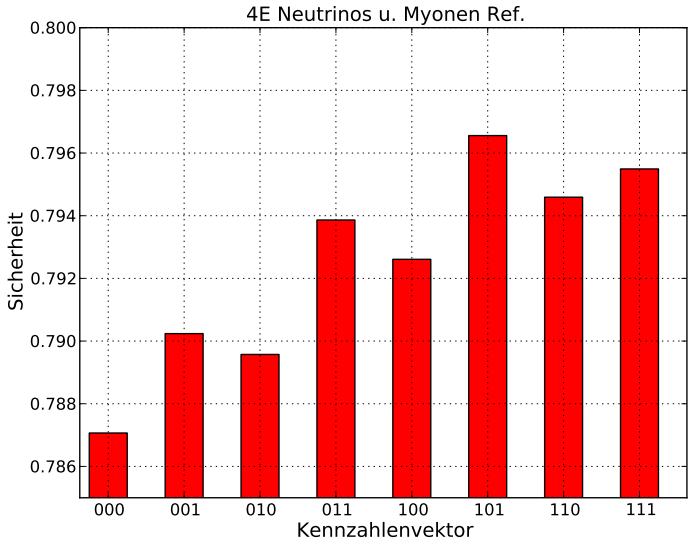

Abbildung A.12: Klassifikationssicherheit bei Klassifizierung 4E für Neutrinos und Myonen

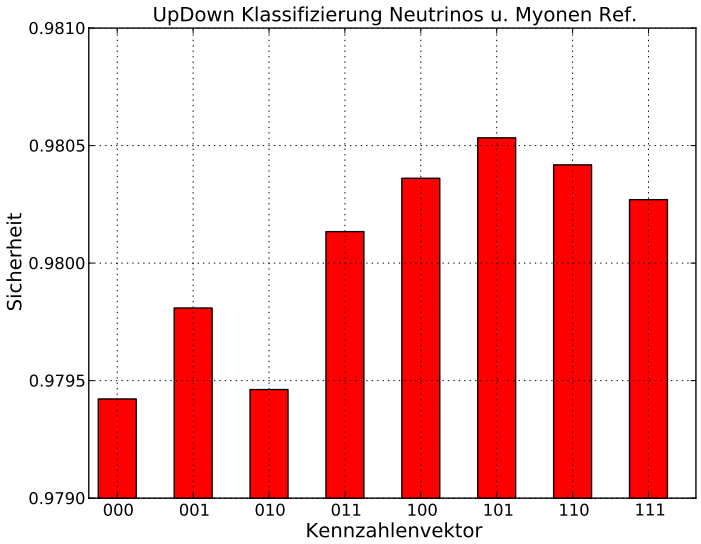

Abbildung A.13: Klassifikationssicherheit für Myonen und Neutrinos bei UpDown Klassifizierung

Anhang B

Liste aller Kennzahlen

In Tabelle B.1 ist der maximale Kennzahlenvektor aufgelistet. Die erste Spalte gibt die Position im Vektor 111 an. Die zweite Spalte gibt Auskunft über die Herkunft der Kennzahl. Hierbei steht „Std“ für Standardkennzahlenvektor, „Egy“ für das Module AntEnergyReco, „Sho“ für DusjShowerReco und „rrF“ für rrFitReco.

Index	Mod	Kennzahlname
1	Std	Number_of_Hits
2	Std	Maximal_OM_charge
3	Std	Level_with_the_maximal_OM_charge
4	Std	Level_diff_Max_to_neighbour_max
5	Std	Time_diff_Max_to_Max_neighb_OM_Max
6	Std	Average_time_diff_Max_to_max_neighb_OM
7	Std	Maximal_time_diff_Max_to_max_neighb_OM
8	Std	Variance_time_diff_Max_to_max_neighb_OM
9	Std	Variance_time_diff_Max_to_max_neighb_OM_norm
10	Std	Time_diff_Max_to_Same_Level
11	Std	Time_diff_Max_to_Same_Level_absolute_Val
12	Std	Time_diff_Max_to_Up_Level
13	Std	Time_diff_Max_to_Up_Level_absolute_Val
14	Std	Time_diff_Max_to_Down_Level
15	Std	Time_diff_Max_to_Down_Level_absolute_Val
16	Std	Time_diff_Max_to_Max_neighb_OM_Max_LONGER
17	Std	Average_time_diff_Max_to_max_neighb_OM_LONGER
18	Std	Maximal_time_diff_Max_to_max_neighb_OM_LONGER
19	Std	Variance_time_diff_Max_to_max_neighb_OM_LONGER
20	Std	Variance_time_diff_Max_to_max_neighb_OM_norm_LONGER
21	Std	Time_diff_Max_to_Same_Level_LONGER

22	Std	Time_diff_Max_to_Same_Level_absolute_Val_LONGER
23	Std	Time_diff_Max_to_Up_Level_LONGER
24	Std	Time_diff_Max_to_Up_Level_absolute_Val_LONGER
25	Std	Time_diff_Max_to_Down_Level_LONGER
26	Std	Time_diff_Max_to_Down_Level_absolute_Val_LONGER
27	Std	Time_diff_Max_to_Max_neighb_OM_Max_LONGER10
28	Std	Average_time_diff_Max_to_max_neighb_OM_LONGER10
29	Std	Maximal_time_diff_Max_to_max_neighb_OM_LONGER10
30	Std	Variance_time_diff_Max_to_max_neighb_OM_LONGER10
31	Std	Variance_time_diff_Max_to_max_neighb_OM_norm_LONGER10
32	Std	Time_diff_Max_to_Same_Level_LONGER10
33	Std	Time_diff_Max_to_Same_Level_absolute_Val_LONGER10
34	Std	Time_diff_Max_to_Up_Level_LONGER10
35	Std	Time_diff_Max_to_Up_Level_absolute_Val_LONGER10
36	Std	Time_diff_Max_to_Down_Level_LONGER10
37	Std	Time_diff_Max_to_Down_Level_absolute_Val_LONGER10
38	Std	Time_diff_Max_to_Max_neighb_OM_Max_LONGER25
39	Std	Average_time_diff_Max_to_max_neighb_OM_LONGER25
40	Std	Maximal_time_diff_Max_to_max_neighb_OM_LONGER25
41	Std	Variance_time_diff_Max_to_max_neighb_OM_LONGER25
42	Std	Variance_time_diff_Max_to_max_neighb_OM_norm_LONGER25
43	Std	Time_diff_Max_to_Same_Level_LONGER25
44	Std	Time_diff_Max_to_Same_Level_absolute_Val_LONGER25
45	Std	Time_diff_Max_to_Up_Level_LONGER25
46	Std	Time_diff_Max_to_Up_Level_absolute_Val_LONGER25
47	Std	Time_diff_Max_to_Down_Level_LONGER25
48	Std	Time_diff_Max_to_Down_Level_absolute_Val_LONGER25
49	Std	Maximal_OM_charge_alltime
50	Std	Level_of_max_OM_charge_alltime
51	Std	Aver_charge_around_the_max_alltime
52	Std	Max_to_Around_charge_ratio_alltime
53	Std	Number_of_OMs_above_line_threshold_alltime
54	Std	Number_of_OMs_above_the_real_threshold_alltime
55	Std	Number_of_lines_above_threshold
56	Std	Size_of_connected_linecluster
57	Std	Level_containing_most_charge
58	Std	Number_of_connected_levels
59	Std	Time_diff_end-start

60	Std	Time_diff_end-start_normiert
61	Std	Time_diff_end-start2
62	Std	Time_diff_end-start2_normiert
63	Std	Time_endsum+startsum
64	Std	Time_endsum+startsum_normiert
65	Std	Time_diff_standard_deviation
66	Std	Time_diff_end-start_sqrtWeighted
67	Std	Time_diff_end-start_sqrtWeigthed_normiert
68	Std	Time_diff_dssum_sqrtWeigthed_normiert
69	Std	Time_diff_desum_sqrtWeigthed_normiert
70	Std	Number_of_Hits
71	Std	Maximal_OM_charge
72	Std	Level_with_the_maximal_OM_charge
73	Std	Level_diff_Max_to_neighbour_max
74	Std	Time_diff_Max_to_Max_neighb_OM_Max
75	Std	Average_time_diff_Max_to_max_neighb_OM
76	Std	Maximal_time_diff_Max_to_max_neighb_OM
77	Std	Variance_time_diff_Max_to_max_neighb_OM
78	Std	Variance_time_diff_Max_to_max_neighb_OM_norm
79	Std	Time_diff_Max_to_Same_Level
80	Std	Time_diff_Max_to_Same_Level_absolute_Val
81	Std	Time_diff_Max_to_Up_Level
82	Std	Time_diff_Max_to_Up_Level_absolute_Val
83	Std	Time_diff_Max_to_Down_Level
84	Std	Time_diff_Max_to_Down_Level_absolute_Val
85	Std	Time_diff_Max_to_Max_neighb_OM_Max_LONGER
86	Std	Average_time_diff_Max_to_max_neighb_OM_LONGER
87	Std	Maximal_time_diff_Max_to_max_neighb_OM_LONGER
88	Std	Variance_time_diff_Max_to_max_neighb_OM_LONGER
89	Std	Variance_time_diff_Max_to_max_neighb_OM_norm_LONGER
90	Std	Time_diff_Max_to_Same_Level_LONGER
91	Std	Time_diff_Max_to_Same_Level_absolute_Val_LONGER
92	Std	Time_diff_Max_to_Up_Level_LONGER
93	Std	Time_diff_Max_to_Up_Level_absolute_Val_LONGER
94	Std	Time_diff_Max_to_Down_Level_LONGER
95	Std	Time_diff_Max_to_Down_Level_absolute_Val_LONGER
96	Std	Time_diff_Max_to_Max_neighb_OM_Max_LONGER10
97	Std	Average_time_diff_Max_to_max_neighb_OM_LONGER10

98	Std	Maximal_time_diff_Max_to_max_neighb_OM_LONGER10
99	Std	Variance_time_diff_Max_to_max_neighb_OM_LONGER10
100	Std	Variance_time_diff_Max_to_max_neighb_OM_norm_LONGER10
101	Std	Time_diff_Max_to_Same_Level_LONGER10
102	Std	Time_diff_Max_to_Same_Level_absolute_Val_LONGER10
103	Std	Time_diff_Max_to_Up_Level_LONGER10
104	Std	Time_diff_Max_to_Up_Level_absolute_Val_LONGER10
105	Std	Time_diff_Max_to_Down_Level_LONGER10
106	Std	Time_diff_Max_to_Down_Level_absolute_Val_LONGER10
107	Std	Time_diff_Max_to_Max_neighb_OM_Max_LONGER25
108	Std	Average_time_diff_Max_to_max_neighb_OM_LONGER25
109	Std	Maximal_time_diff_Max_to_max_neighb_OM_LONGER25
110	Std	Variance_time_diff_Max_to_max_neighb_OM_LONGER25
111	Std	Variance_time_diff_Max_to_max_neighb_OM_norm_LONGER25
112	Std	Time_diff_Max_to_Same_Level_LONGER25
113	Std	Time_diff_Max_to_Same_Level_absolute_Val_LONGER25
114	Std	Time_diff_Max_to_Up_Level_LONGER25
115	Std	Time_diff_Max_to_Up_Level_absolute_Val_LONGER25
116	Std	Time_diff_Max_to_Down_Level_LONGER25
117	Std	Time_diff_Max_to_Down_Level_absolute_Val_LONGER25
118	Std	Maximal_OM_charge_alltime
119	Std	Level_of_max_OM_charge_alltime
120	Std	Aver_charge_around_the_max_alltime
121	Std	Max_to_Around_charge_ratio_alltime
122	Std	Number_of_OMs_above_line_threshold_alltime
123	Std	Number_of_OMs_above_the_real_threshold_alltime
124	Std	Number_of_lines_above_threshold
125	Std	Size_of_connected_linecluster
126	Std	Level_containing_most_charge
127	Std	Number_of_connected_levels
128	Std	Time_diff_end-start
129	Std	Time_diff_end-start_normiert
130	Std	Time_diff_end-start2
131	Std	Time_diff_end-start2_normiert
132	Std	Time_endsum+startsum
133	Std	Time_endsum+startsum_normiert
134	Std	Time_diff_standard_deviation
135	Std	Time_diff_end-start_sqrtWeighted

136	Std	Time_diff_end-start_sqrtWeigthed_normiert
137	Std	Time_diff_dssum_sqrtWeigthed_normiert
138	Std	Time_diff_desum_sqrtWeigthed_normiert
139	Egy	energy
140	Egy	P0
141	Egy	P1
142	Egy	P2
143	Egy	P3
144	Egy	P4
145	Egy	P5
146	Egy	P6
147	Egy	P7
148	Egy	P8
149	Egy	P9
150	Egy	P10
151	Egy	P11
152	Egy	P12
153	Egy	P13
154	Egy	P14
155	Egy	P15
156	Egy	P16
157	Egy	P17
158	Egy	P18
159	Egy	P19
160	Egy	P20
161	Sho	DusjShowerRecoFinalFitDegreesOfFreedom
162	Sho	DusjShowerRecoFinalFitLogLikelihood
163	Sho	DusjShowerRecoFinalFitMinimizerCalls
164	Sho	DusjShowerRecoFinalFitReducedLogLikelihood
165	Sho	DusjShowerRecoVertexFitDegreesOfFreedom
166	Sho	DusjShowerRecoVertexFitLogLikelihood
167	Sho	DusjShowerRecoVertexFitMinimizerCalls
168	Sho	DusjShowerRecoVertexFitReducedLogLikelihood
169	Sho	FitConvergencePositionAzimuth
170	Sho	FitConvergencePositionEnergy
171	Sho	FitConvergencePositionTime
172	Sho	FitConvergencePositionX
173	Sho	FitConvergencePositionY

174	Sho	FitConvergencePositionZ
175	Sho	FitConvergencePositionZenith
176	Sho	FitHorizontalDistanceToDetectorCenter
177	Sho	FitNumberOfStrings
178	Sho	FitQuadrupoleMoment
179	Sho	FitTimeResidualChiSquare
180	Sho	FitTotalCharge
181	Sho	FitVerticalDistanceToDetectorCenter
182	Sho	ShowerIdentifierHorizontalDistanceToDetectorCenter
183	Sho	ShowerIdentifierReducedChiSquare
184	Sho	ShowerIdentifierVerticalDistanceToDetectorCenter
185	rrF	rrFitSlope
186	Std	aaZenPos
187	Std	aaAzPos
188	Std	aaLamdaPos
189	Std	aaBetaPos

Tabelle B.1: Auflistung aller Kennzahlen

Anhang C

Graphen zu den optimierten Kennzahlenvektoren

Bei den Graphen zu den optimierten Kennzahlenvektoren wurde die berechnete Sicherheit gegen die Anzahl der verwendeten Kennzahlen aufgetragen. Zusätzlich werden die Nummern der Kennzahlen angegeben welche das lokale Sicherheitsmaximum ergeben (Tabelle C.1). In der letzten Spalte, „beste Kombination“, steht nur ein Eintrag wenn der Kennzahlenvektor, der bei der Suche nach den optimierten Vektoren das beste Ergebnis geliefert hat, nicht alle sechs optimierten Kennzahlen benutzt. In der letzten Spalte stehen dann die Kennzahlen, die die höchste Sicherheit ermöglichten.

Klassifikation	Teilchen	optimierter Kennzahlenvektor	beste Kombination
0E	Myonen	141, 68, 135, 18, 79, 136	141, 68, 135
1E	Myonen	141, 3, 1, 185, 139, 150	-
0E	Neutrinos	139, 49, 132, 185, 159, 70	-
1E	Neutrinos	139, 185, 159, 123, 17, 26	-
2E	Neutrinos	1, 139, 49, 185, 188, 185	-
3E	Neutrinos	1, 139, 52, 183, 188, 185	-
4E	Neutrinos	53, 139, 188, 174, 52, 84	-
0E	Neutrinos u. Myonen	141, 128, 139, 159, 103, 186	-
1E	Neutrinos u. Myonen	70, 129, 159, 139, 156, 35	-
2E	Neutrinos u. Myonen	128, 139, 121, 52, 186, 12	-
3E	Neutrinos u. Myonen	141, 129, 139, 52, 51, 57	-
4E	Neutrinos u. Myonen	70, 87, 18, 144, 18, 70	70, 87, 18, 144, 18
UD	Neutrinos u. Myonen	128, 103, 186, 188, 121, 60	-

Tabelle C.1: Auflistung der optimierten Kennzahlenvektoren

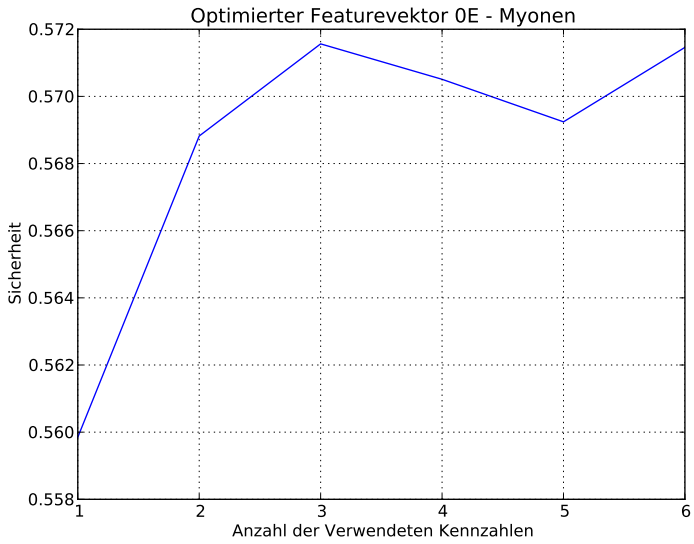

Abbildung C.1: Klassifikationssicherheiten unter Verwendung von optimierten Kennzahlen der Klassifikation 0E für Myonen

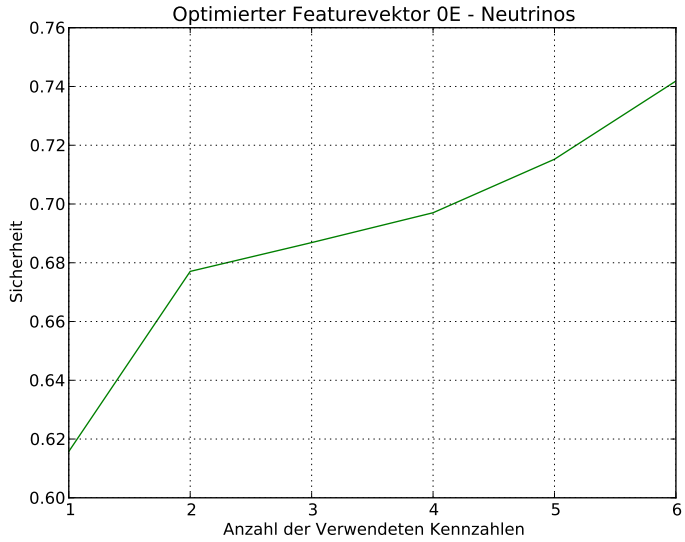

Abbildung C.2: Klassifikationssicherheiten unter Verwendung von optimierten Kennzahlen der Klassifikation 0E für Neutrinos

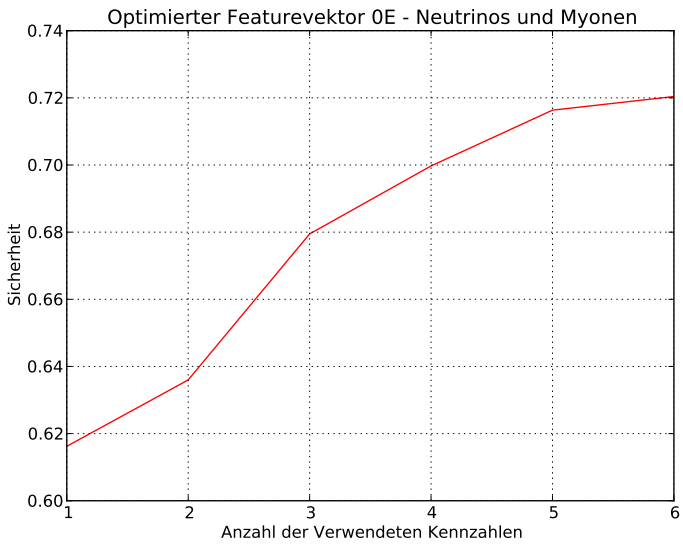

Abbildung C.3: Klassifikationssicherheiten unter Verwendung von optimierten Kennzahlen der Klassifikation 0E für Myonen und Neutrinos

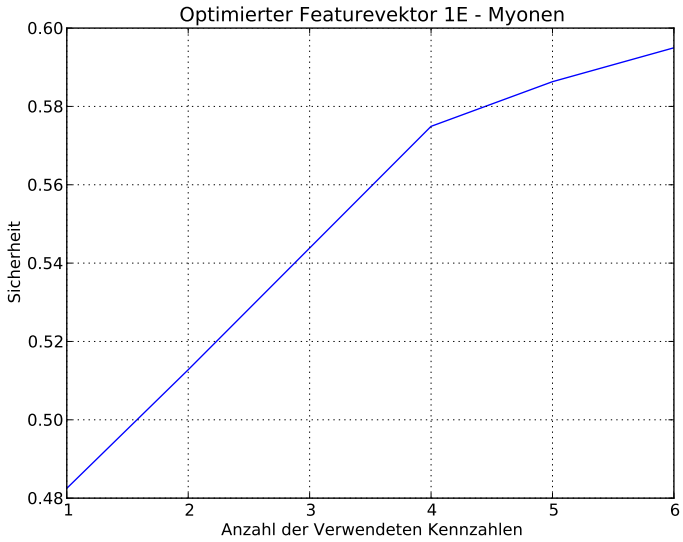

Abbildung C.4: Klassifikationssicherheiten unter Verwendung von optimierten Kennzahlen der Klassifikation 1E für Myonen

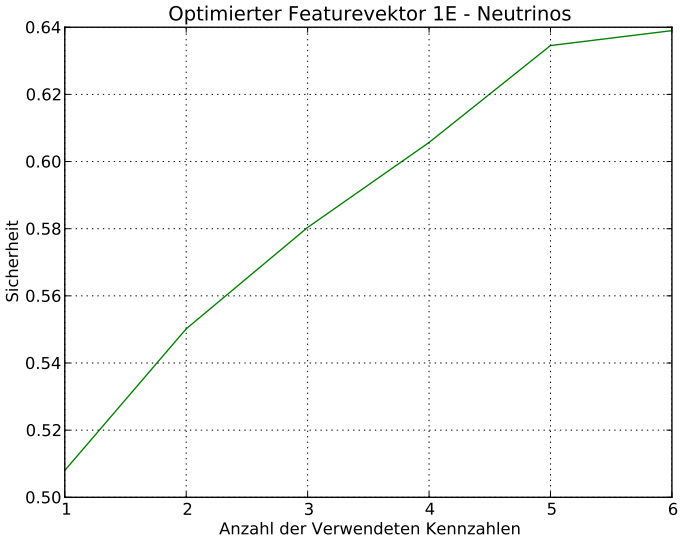

Abbildung C.5: Klassifikationssicherheiten unter Verwendung von optimierten Kennzahlen der Klassifikation 1E für Neutrinos

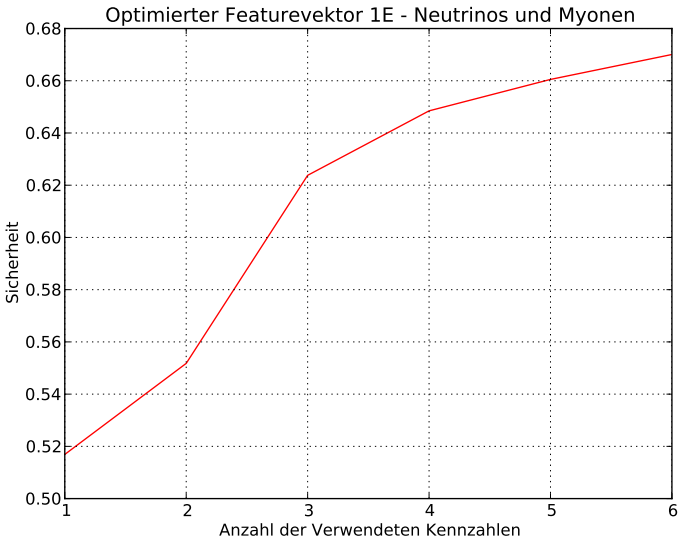

Abbildung C.6: Klassifikationssicherheiten unter Verwendung von optimierten Kennzahlen der Klassifikation 1E für Myonen und Neutrinos

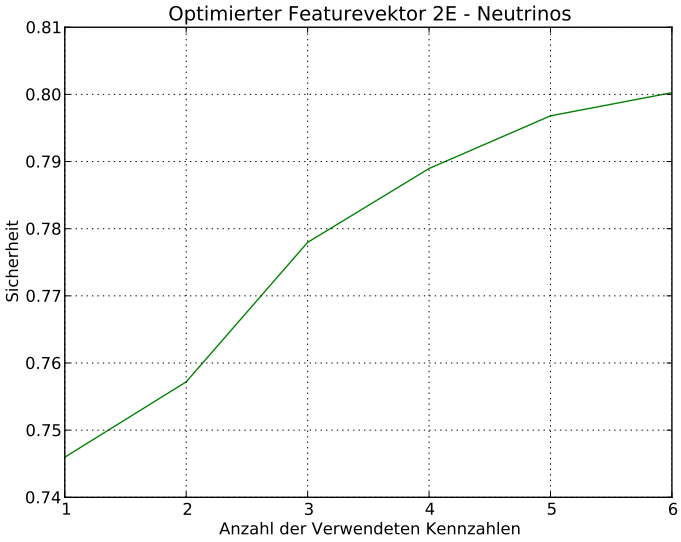

Abbildung C.7: Klassifikationssicherheiten unter Verwendung von optimierten Kennzahlen der Klassifikation 2E für Neutrinos

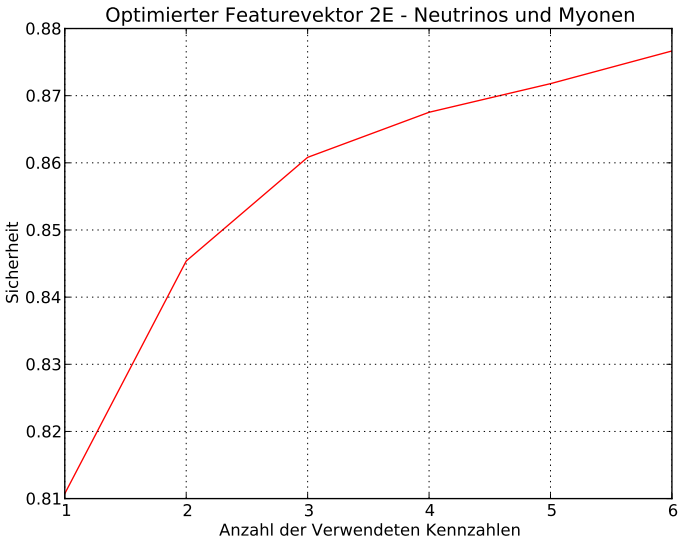

Abbildung C.8: Klassifikationssicherheiten unter Verwendung von optimierten Kennzahlen der Klassifikation 2E für Myonen und Neutrinos

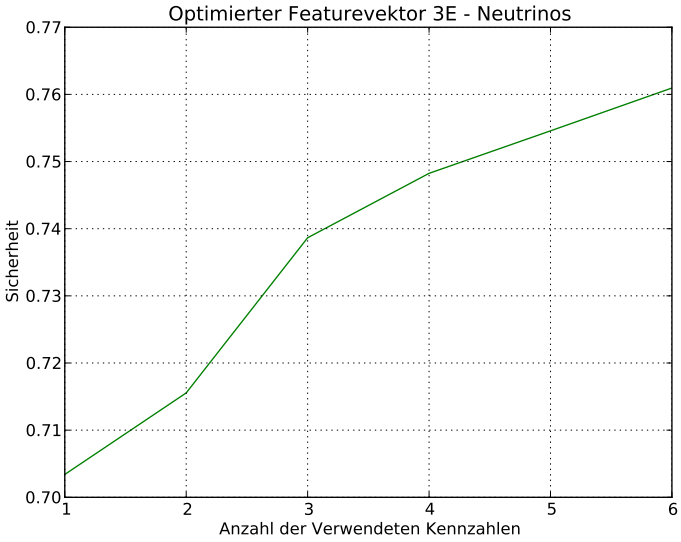

Abbildung C.9: Klassifikationssicherheiten unter Verwendung von optimierten Kennzahlen der Klassifikation 3E für Neutrinos

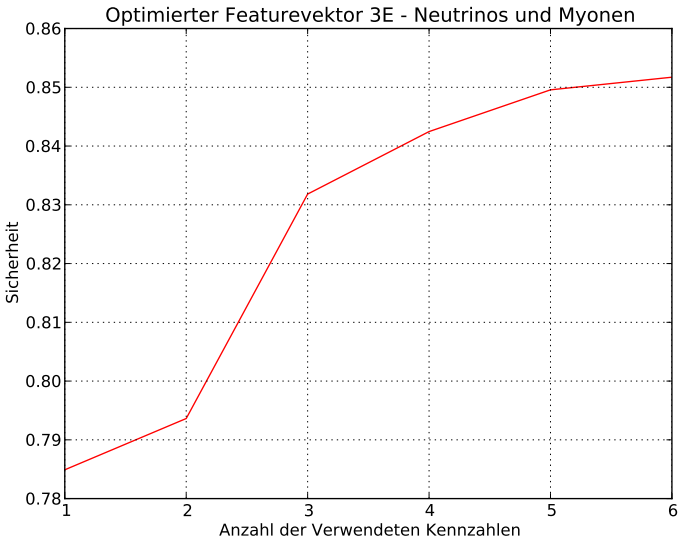

Abbildung C.10: Klassifikationssicherheiten unter Verwendung von optimierten Kennzahlen der Klassifikation 3E für Myonen und Neutrinos

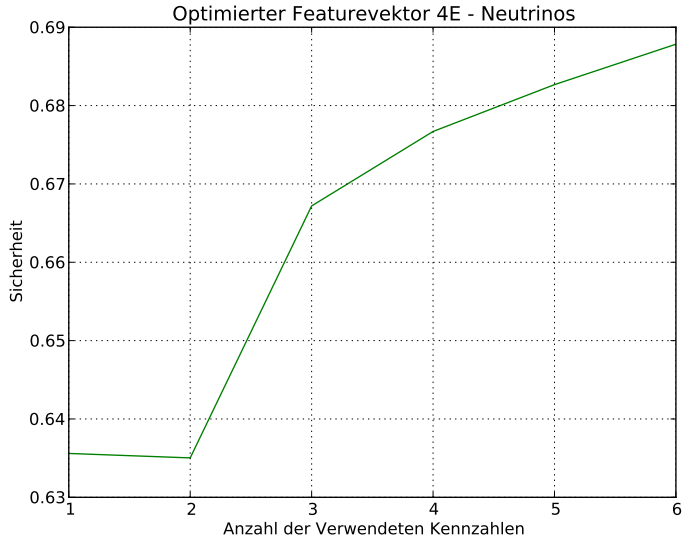

Abbildung C.11: Klassifikationssicherheiten unter Verwendung von optimierten Kennzahlen der Klassifikation 4E für Neutrinos

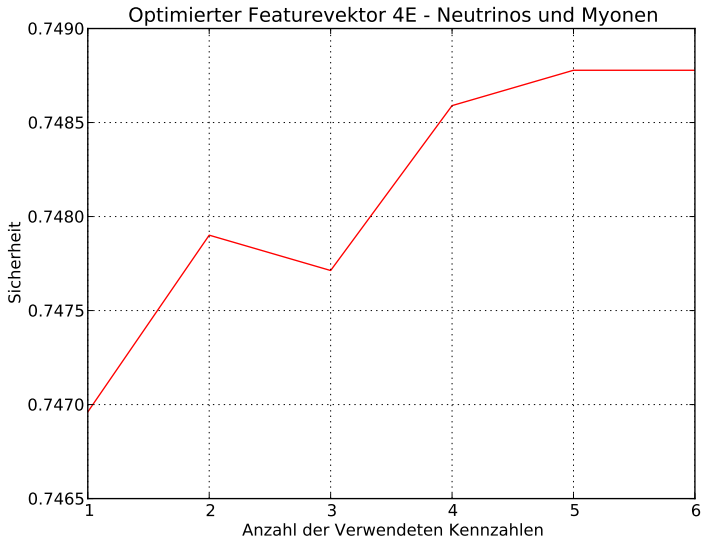

Abbildung C.12: Klassifikationssicherheiten unter Verwendung von optimierten Kennzahlen der Klassifikation 4E für Myonen und Neutrinos

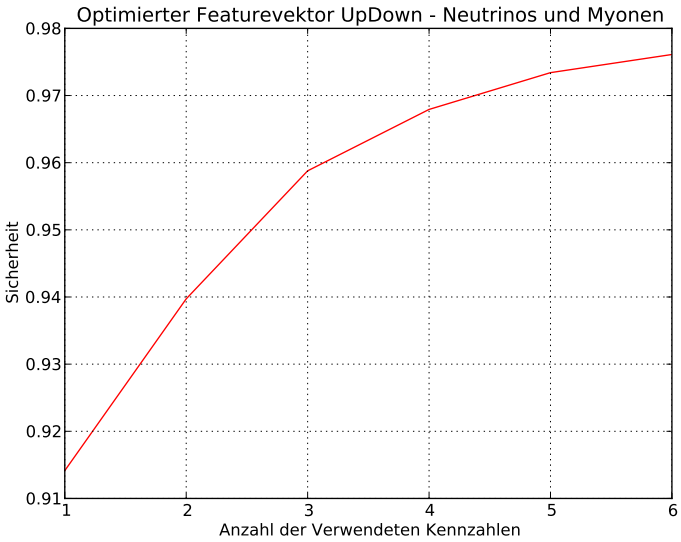

Abbildung C.13: Klassifikationssicherheiten unter Verwendung von optimierten Kennzahlen der UpDown Klassifikation für Myonen und Neutrinos

Anhang D

Energiespektren der Simulationen

Die im Folgenden abgebildeten Graphen stellen das Energiespektrum der simulierten Events dar. Die Breite der verwendeten Bins ist bei allen Graphen je $100\,GeV$.

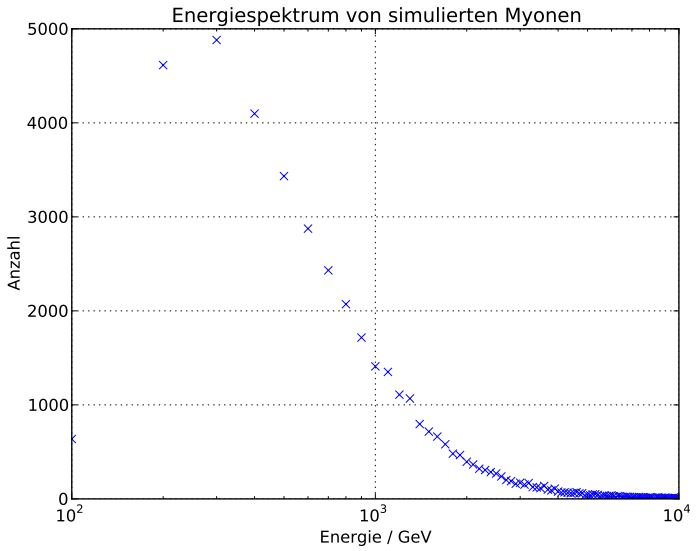

Abbildung D.1: Energiespektrum der simulierten Myonen, niederenergetischer Teil

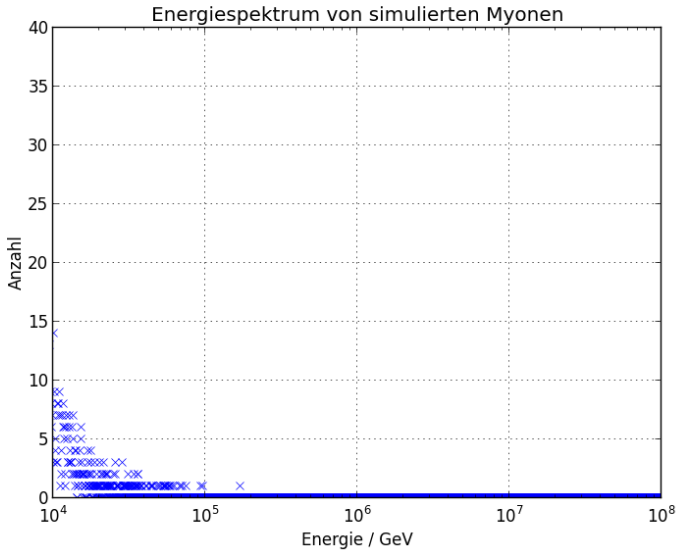

Abbildung D.2: Energiespektrum der simulierten Myonen, hochenergetischer Teil

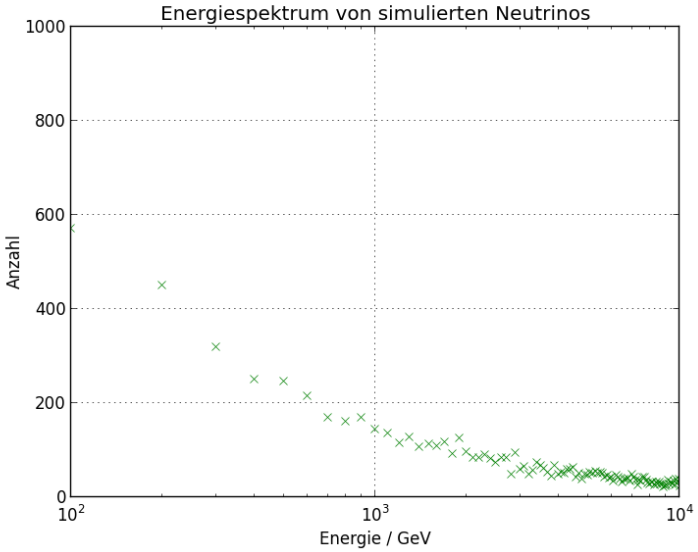

Abbildung D.3: Energiespektrum der simulierten Neutrinos, niederenergetischer Teil

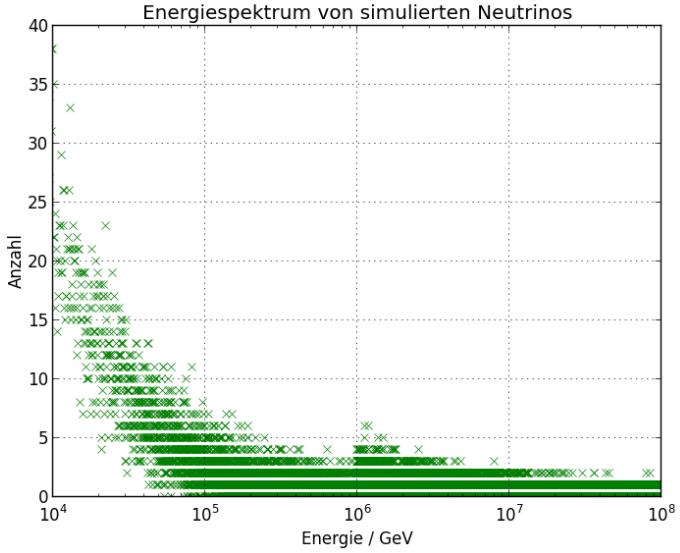

Abbildung D.4: Energiespektrum der simulierten Neutrinos, hochenergetischer Teil

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[1] ANTARES home page, 2012. http://antares.in2p3.fr/.

[2] R.E. Bellman. *Adaptive Control Process.* Pincton University Press, 1961.

[3] B.R.Martin. *Nuclear and Particle Physics.* John Wiley and Sons, Inc., 2009.

[4] Alexander Enzenhöfer. Quellen hochenergetischer Neutrinos und Neutrinoteleskope. Presentation, 2008.

[5] Dietrich W. R. Paulus et al. *Applied Pattern Recognition.* Vieweg Verlag, 2003.

[6] Hannu Karttunen et al. *Fundamental Astronomy.* Springer Verlag, 2006.

[7] K. Nakamura et al. *Particle Physics Booklet.* IOP Publishing, 2010.

[8] M. Villar-Martin et al. Kinematically quiet halos around z 2.5 radio galaxies. Keck spectroscopy. arxiv:0309012v1, 2008.

[9] S. Adrian Martines et al. Search for Cosmic Neutrino Point Sources with Four Years of Data from the ANTARES Telescope. arXiv:1207.3105v2, 2012.

[10] Stefan Geißelsoeder. Classification of events for the ANTARES neutrino detector. Master's thesis, Erlangen Centre for Astroparticle Physics, 2010.

[11] Tin Kam Ho. Random Decision Forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:832–844, 1998.

[12] Marc L. Kutner. *Astronomy, A Physical Perspective.* Cambridge University Press, 2003.

[13] Gabriela Pavalas. Search for Nuclearites with the ANTARES detector. arxiv:1010.2071v1, 2010.

[14] M.Spurio T.Chiarusi. High-energy astrophysics with neutrino telescope. Technical report, Dipartimento di Fisica, Universita di Bologna; INFN, Sezione di Bologna, 2009.