

EMPA Materials Science and Technology
Berufsmaturitätsschule Zürich

Wettbewerbsarbeit für Schweizer Jugend forscht

KUNSTSTOFFTRENNUNG MIT TERAHERTZSTRAHLUNG

L. Nicklaus Cáceres, Sofie L. Gnannt
Zürich, 26.03.2023

Abstract

In diesem Projekt werden die Möglichkeiten der Terahertz (THz) Time-Domain-Spektroskopie (TDS) als Methode zur zuverlässigen Klassifizierung von Kunststoffen für das Recycling untersucht. Polymere sind für THz-Strahlung relativ transparent, was uns erlaubt, sie in Transmission mit einem THz Time-Domain-Spektrometer zu untersuchen. Dadurch lernen wir, wie ein spezifisches Material mit der Strahlung interagiert und können die unterschiedlichen Kunststoffe voneinander unterscheiden.

Mit Hilfe eines etablierten Rechenalgorithmus für die Extraktion des Brechungsindex und der Absorption sowie von Modellen des maschinellen Lernens zeigen wir, dass es möglich ist, Kunststoffabfälle automatisiert und zuverlässig in vier der wichtigsten wiederverwertbaren Klassen zu kategorisieren.

Inhaltsverzeichnis

1 Vorwort	2
2 Ausgangslage	3
2.1 Idee	3
2.2 Projektziel	3
3 Hintergrund	4
3.1 Begriffe	4
3.2 Kunststoffe	4
3.2.1 Polypropylen (PP)	4
3.2.2 Polystyrol (PS)	4
3.2.3 Polyethylenterephthalat (PET)	4
3.2.4 Polyvinylchlorid (PVC)	5
3.3 THz-Strahlung	5
3.4 THz Time Domain Spectroscopy	5
3.4.1 Messaufbau	5
3.4.2 THz-TDS mit Kunststoff	7
3.5 Maschinelles Lernen	8
3.5.1 Einführung in maschinelle Lerntechniken	8
3.5.2 Training und Schätzung der Vorhersageleistung	8
3.5.3 Merkmalsauswahl/Reduzierung der Dimensionalität	9
3.5.4 Hyperebenen	9
3.5.5 Kernel-basierte Methoden	II
3.6 Support Vector Machines (SVMs)	II
3.6.1 Mehrklassen-SVMs	II
4 Arbeitsschritte	13
4.1 Probenbeschaffung	13
4.2 Messungen und Auswertung	14
4.2.1 Extraktion von Brechungsindex und Absorption	14
4.2.2 Merkmalsextraktion mit Python	16
4.2.3 Anwendung von SVMs	16
4.3 Agile Robots for Tomorrow's Lab (ARTLab)	18
5 Ergebnisse	19
6 Diskussion	20
6.1 Bestimmung der Probendicke	20
6.2 Reflexionsmessung	21
6.3 Verzerrung der Stichprobe und Verallgemeinerungsfähigkeit	21
6.4 Informationsverlust durch Merkmalsauswahl	23
6.5 Exploration in der industriellen Anwendung	24
7 Fazit	25
8 Danksagung	25
9 Ausblick	26
9.1 Mögliche Weiterentwicklung	26

I Vorwort

Der nationale Wettbewerb der Schweizer Jugend Forscht (SJF) ist für uns eine gute Gelegenheit, unser Wissen und unsere Fähigkeiten zu präsentieren und weiterzuentwickeln. Als Physiklaboranten bilden wir eine Brücke zwischen Theorie und Anwendung. Wir wollen zeigen, wie ein interessantes physikalisches Phänomen in eine Lösung für ein gesellschaftliches Problem umgesetzt werden kann.

Weil unser Lehrbetrieb als Forschungsinstitut für Materialwissenschaften und Technologie für Lösungen zu gesellschaftlichen Herausforderungen sowie für die Nachhaltigkeit steht, wollen wir mit unserem Projekt gewissermassen unseren Betrieb widerspiegeln. Das Recyceln von Kunststoffen ist ein bedeutendes Thema in unserer Zeit. Es ist wichtig, dass wir gute Möglichkeiten entwickeln, Materialien zu trennen, bevor man sie recyceln und später wiederverwenden kann. Maschinelles Lernen wird von Tag zu Tag wichtiger. Es ist eine Technologie mit viel Potenzial, Ereignisse und Beobachtungen kategorisieren zu können. Genau diese zwei Themen haben wir in unserem Projekt kombiniert.

Da wir zuvor schon mit dem Terahertz-Spektrometer gearbeitet haben, kam uns die Idee ziemlich schnell, diesen in unser Projekt der Kunststofftrennung zu integrieren. Maschinelles Lernen kann gebraucht werden, um unterschiedliche Objekte mit verschiedenen Eigenschaften zu unterscheiden. Durch Anwenden von Methoden des maschinellen Lernens werden wir aus Terahertz-Spektren von Kunststoffproben charakteristische Eigenschaften zur Bestimmung des Kunststoffes erarbeiten. In einem weiteren Schritt wollen wir mit Hilfe der Robotertechnologie einen möglichen Ansatz für die Skalierung in Recyclingbetrieben aufzeigen.

2 Ausgangslage

2.1 Idee

Recycling ist komplex. Die zu rezyklierenden Produkte müssen nicht nur den Prozess der physikalischen oder chemischen Umwandlung von Abfällen in wiederverwendbare Produkte durchlaufen. Es muss zu Beginn auch sichergestellt werden, dass es sich um dieselbe Art von Werkstoff handelt, damit Verunreinigungen und unerwünschte Materialeigenschaften vermieden werden können. Heutzutage stützt sich das Klassifizierungsverfahren für Polymere auf physikalische Merkmale wie Dichte, magnetische oder elektrische Eigenschaften. Viele davon haben Schwierigkeiten, Polymere mit sehr ähnlichen Merkmalen zu klassifizieren, so dass es notwendig ist, mehr als nur eine dieser Methoden anzuwenden [Serranti and Bonifazi, 2019].

Die THz-Absorption in Polymeren ist durch deren Morphologie gegeben und wird auf die makromolekulare Struktur zurückgeführt [Wietzke et al., 2011]. Viele Materialien weisen im THz Spektrum einzigartige Absorptionslinien auf, die auch "Fingerabdrücke" genannt werden. Anhand dieser Absorptionslinien lässt sich auf die vorhandenen chemischen Bindungen schliessen und wir sollten in der Lage sein, zwischen den Hauptkategorien umweltrelevanter Kunststoffe zu unterscheiden, die Polyethylenterephthalat (PET), Polypropylen (PP), Polyvinylchlorid (PVC) und Polystyrol (PS) umfassen, aber nicht darauf beschränkt sind. Maschinelles Lernen ermöglicht die schnelle Klassifizierung von Daten in mehrere Kategorien aufgrund von Schlüsselmerkmalen, die in einigen Fällen mit einer ausreichend grossen Trainingsmenge durch Algorithmen zur Dimensionalitätsreduktion extrahiert werden können [Bishop, 2006]. Unsere Idee ist, mit Hilfe der einzigartigen Absorptionseigenschaften im THz Spektrum, und mittels einem Algorithmus für maschinelles Lernen, die meisten Kunststoffe des täglichen Gebrauchs in ihre entsprechenden Kategorien für das Recycling einteilen zu können.

2.2 Projektziel

Unser Ziel ist es, die THz Time Domain Spectroscopy (THz-TDS) als zuverlässige Methode zur Gewinnung makromolekularer Informationen aus unbekannten Polymeren vorzustellen, um sie im Recycling Prozess zur Materialklassifizierung einzusetzen. Zu diesem Zweck möchten wir ein maschinelles Lernmodell erstellen, das mit Daten aus alltäglichen Kunststoffen trainiert wird. Wir möchten uns auf Kunststoffabfälle konzentrieren, die typischerweise im Haushalt anfallen, und in der Lage sein, auf die handelsüblichen Recyclingkategorien zu verallgemeinern. Außerdem möchten wir die Möglichkeiten der Automatisierung unserer Methode durch den Einsatz von Kleinrobotern als Hilfsmittel für unsere Messungen untersuchen. Das zweite Ziel unseres Projekts ist es, unseren Beruf und das breite Spektrum an Wissen, das für die Arbeit als Physiklaborant erforderlich ist, zu präsentieren. Wir möchten jungen Sekundarschülern die vielen interessanten Möglichkeiten aufzeigen, wie das wissenschaftliche Arbeiten und mathematische Algorithmen die reale Welt beeinflussen können, in dem sie uns anwendbare Lösungen liefern.

3 Hintergrund

3.1 Begriffe

- **Brechungsindex (n):** Der Brechungsindex ist eine optische Materialeigenschaft und ist im Allgemeinen komplexwertig, besteht also aus einem Realteil und einem Imaginärteil. Der Realteil bestimmt wie schnell sich das Licht im Medium ausbreitet. Der Imaginärteil beschreibt wie stark das Licht vom Material absorbiert wird.
- **Absorption:** Die Absorption elektromagnetischer Strahlung ist die Fähigkeit der Materie, die Energie eines Photons aufzunehmen.
- **Extinktionskoeffizient (κ):** Der Extinktionskoeffizient und der Absorptionskoeffizient (α) stehen in folgendem Bezug zueinander: $\alpha = \frac{4\pi f \kappa}{c}$, wobei f die Frequenz und c die Lichtgeschwindigkeit ist.
- **Thermoplaste:** Kunststoffe, die sich bei einem bestimmten Temperaturbereich verformen lassen.
- **Polymer:** Ein chemischer Stoff, der aus Monomeren aufgebaut ist. Monomere (z. B. Ethen C_2H_4) sind niedermolekulare, reaktionsfähige Moleküle.
- **Polare Stoffe:** Ein polarer Stoff besteht aus polaren Molekülen, welche sich durch ein permanentes elektrisches Dipolmoment auszeichnen. Unpolare Moleküle haben kein oder ein sehr kleines Dipolmoment.
- **Fast Fourier Transform:** Die Fast Fourier Transformation ist eine Implementierung der Fourier Transformation, die wenig Rechenleistung benötigt. Mithilfe der FFT können wir Signale im Zeit/Orts-Raum in den Frequenzraum umrechnen.

3.2 Kunststoffe

3.2.1 Polypropylen (PP)

Polypropylen, abgekürzt PP ist ein thermoplastisches Polymer mit der Summenformel $(C_3H_6)_n$. PP ist ein weisses, mechanisch robustes Material mit einer hohen chemischen Beständigkeit. Polypropylen ist bei Raumtemperatur beständig gegen fast alle organischen Lösungsmittel und Fette. Bei erhöhten Temperaturen können einige unpolare Lösungsmittel PP auflösen. Polypropylen hat eine geringe Dichte, zwischen 0.895 und 0.92, ist normalerweise zäh und flexibel und weist eine gute Ermüdbeständigkeit auf. Polypropylen wird häufig für Verpackungen verwendet z.B. für Reinigungsprodukte, Klebefolien und Trinkhalme aber auch Flaschenverschlüsse, Innenausstattung für Maschinen- und Fahrzeubau und noch vielem mehr [Wikipedia contributors, 2022b]. Mit einem globalen Marktvolumen von etwa 74 Mio. t im Jahr 2018 ist Polypropylen der meistverwendete Kunststoff weltweit [Braun et al., 2019].

Der Brechungsindex von Polypropylen bei 2 THz ist ungefähr 1.5 mit einer Streuung von weniger als 0.5%. Oberhalb von 2 THz hat es mehrere starke Absorptionsbänder [Bründermann et al., 2012].

3.2.2 Polystyrol (PS)

Polystyrol, abgekürzt PS, ist ein synthetisches Polymer mit der Summenformel $(C_8H_8)_n$. Polystyrol kommt in fester und geschäumer Form vor. PS ist wasserbeständig und widerstandsfest gegen viele Säuren und Basen, wird aber von Lösungsmitteln, wie zum Beispiel Aceton, leicht angegriffen. Der herkömmliche Allzweck-Polystyrol ist hart, spröde und transparent, er lässt sich auch einfärben. Als thermoplastisches Polymer befindet sich Polystyrol bei Raumtemperatur im festen Zustand, sobald das Polymer aber über 100°C erhitzt wird, wird er weich und lässt sich verformen. Es werden jährlich mehrere Millionen Tonnen PS produziert, in der Regel durch Termoformen oder Spritzguss. Polystyrol wird zur Herstellung von Einweg-Plastikbesteck und -geschirr, CD-Hüllen, Sensorgehäuse, Styropor und vielem mehr verwendet. PS wird dort überall eingesetzt, wo ein steifer kostengünstiger Kunststoff gefragt ist [Wikipedia contributors, 2022c]. Polystyrol hat einen hohen Brechungsindex 2.08 ± 0.01 im THz Spektrum, der dem von Quarz sehr ähnlich ist. Der Absorptionskoeffizient steigt mit zunehmender Frequenz von 0 bis 2.4 THz [Bründermann et al., 2012].

3.2.3 Polyethylenterephthalat (PET)

Polyethylenterephthalat, bekannter als PET, mit der Summenformel $(C_{10}H_8O_4)_n$, gehört zu der Familie der Polyester. 2008 lag die PET Produktion bei 40 Millionen Tonnen. PET ist bei Raumtemperatur fest, ab 70°C wird PET weicher und lässt sich verformen. Polyethylenterephthalat ist beständig gegen viele Chemikalien, daher kommt auch, dass viele Lebensmittel in PET verpackt sind. Wegen genau dieser Beständigkeit sind auch viele Verpackungen im Labor und der Medizin aus PET. PET ist gegenüber stärkeren anorganischen Säuren (z.B. Schwefelsäure, Salzsäure) jedoch nicht beständig. PET hat eine hohe Bruchfestigkeit und Verformbeständigkeit bei über 80°C, auch ein gutes Gleit- und Verschweissverhalten kann PET aufweisen. Jedoch ist die Schlagzähigkeit gering [Wikipedia, 2022a].

Der Brechungsindex von PET im THz Bereich ist ungefähr 1.72, der Absorptionskoeffizient steigt mit zunehmender Frequenz von 1.5 bis 10 THz, mit einer augeprägten Resonanz bei 4.2 THz [Bründermann et al., 2012].

3.2.4 Polyvinylchlorid (PVC)

Polyvinylchlorid, abgekürzt PVC, ist ein thermoplastisches Polymer mit der Summenformel $(C_2H_3Cl)_n$. PVC Kunststoffe werden in Hart- und Weich-PVC unterteilt. Für die Herstellung von Schallplatten, Fensterprofilen und Rohren wird ein Hart-PVC verwendet. Weich-PVC wird für Kabelummantelungen, Schuhsohlen, Schläuche und Bodenbeläge verwendet. Weich-PVC enthält bis zu 40% Weichmacher. Das Zugeben von Weichmacher verleiht dem Polymer plastische Eigenschaften, dieser lagert sich bei der thermoplastischen Verarbeitung zwischen den Molekülketten des PVC ein. Dadurch lockert der Weichmacher das Gefüge des PVC. Polyvinylchlorid lässt sich lange halten, nimmt kaum Wasser auf und zerstört sich durch Sonnenlicht nicht. PVC lässt sich einfärben und ist sehr preisgünstig. Polyvinylchlorid ist gegen einige Säuren und Laugen beständig. Angegriffen wird PVC unter anderem durch Aceton, Benzol und konzentrierte Salzsäure. Hart-PVC lässt sich gut, Weich-PVC schlecht spanabhebend verarbeiten. Bei Temperaturen von 120 bis 150 °C lässt sich PVC spanlos verformen [Wikipedia, 2022b].

PVC hat bei 2 THz einen Brechungsindex von 1.6, der mit steigendem Weichmachergehalt abnimmt [Sommer et al., 2018].

3.3 THz-Strahlung

THz-Strahlung ist eine elektromagnetische Welle und liegt zwischen der Infrarotstrahlung und den Mikrowellen. Die THz-Strahlung liegt im Frequenzbereich von 0,3 bis 10 THz und entspricht einer Wellenlänge von $30\mu\text{m}$ bis 1mm , obwohl die Grenzen nicht von allen Quellen gleich definiert werden. THz-Strahlung ist aufgrund ihrer geringen Photonenergie nicht ionisierend. Sie wird von Metallen reflektiert und von polaren Stoffen (z.B. Wasser) absorbiert. Dielektrischen Materialien, die elektrisch schwach oder gar nicht leiten, wie z. B. Kunststoff reflektieren und absorbieren in der Regel weniger und weisen eine hohe Transmission auf. THz-Strahlung ist ein sehr interessantes Forschungsbereich, da man erst seit etwa 15 Jahren leistungsfähige Strahler und Detektoren für diesen Spektralbereich hat. Die Sensoren sind aber auch heute noch ein Problem, da die THz-Strahlung in dem Spektralbereich liegt, den Überlagerungsempfänger, wie Radio- und Fernsehempfänger, fast nicht mehr und optische Sensoren noch nicht abdecken. Man hatte relativ lange bis THz-Strahlung erzeugt werden konnte, deswegen sprach man früher von der Terahertz-Lücke. Das Problem ist aber noch lange nicht gelöst, die Sender mit genügend Ausgangsleistung sind wegen geringen Stückzahlen noch sehr teuer, und die Empfänger sollten noch schwächere Signale empfangen können. Wie bereits oben erwähnt absorbieren polare Stoffe THz-Strahlung. Unter anderem absorbieren Wasser und Luftfeuchtigkeit die THz-Strahlung und schwächen so das Signal ab. THz-Strahlung hat schon einige Anwendung im alltäglichen Leben gefunden, z. B. in der zerstörungsfreien Prüfung, durch spektroskopische Informationen, die räumlich aufgelöst werden. Dadurch werden Defekte im Körperinneren sichtbar. Auch in der Medizintechnik kann THz-Strahlung zum Einsatz kommen, da THz-Strahlung im Gegensatz zu Röntgenstrahlung keine ionisierende Strahlung ist, werden weder lebendes Gewebe noch DNA beschädigt. In Gewebe mit geringem Wassergehalt können einige Frequenzen einige Millimeter weit eindringen. Auch in der Zahntechnik kann THz-Strahlung zum Einsatz kommen. In der Sicherheitstechnik könnte man versteckte Waffen von Personen schon aus der Ferne erkennen. Das Potential für Forschung und neue Technologien ist immer noch sehr gross. [Brüdermann et al., 2012].

3.4 THz Time Domain Spectroscopy

3.4.1 Messaufbau

Der Aufbau eines Transmissions THz Time Domain Spectrometer (THz-TDS) ist in Abbildung 1 dargestellt. Für die

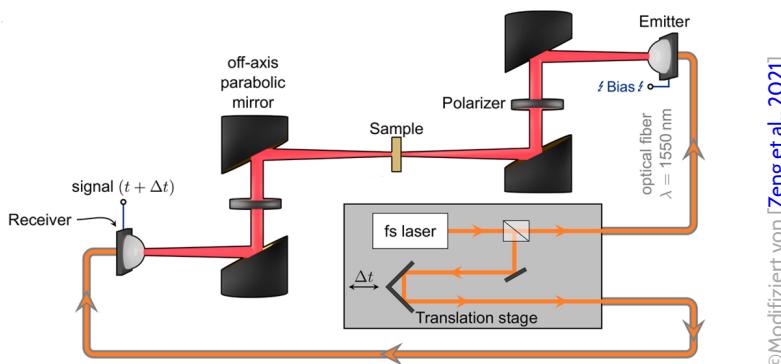


ABBILDUNG 1 – Unser THz-TDS Aufbau an der Empa

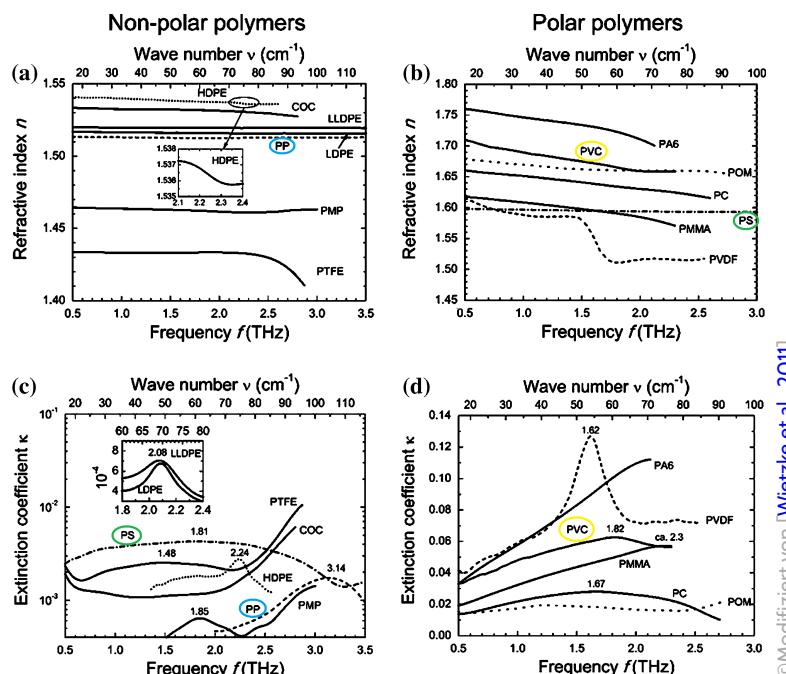
Erzeugung von gepulster THz-Strahlung wird ein Femtosekundenlaser (fs laser) verwendet, der Laserpuls wird in zwei Strahlen geteilt. Der erste Strahl wandert über eine Glasfaser (optical fiber) zum Detektor (Receiver), der andere Strahl propagiert in einer zweiten Glasfaser zum Sender (Emitter). Der Laserpuls erzeugt in dem Sender Ladungsträger, diese

werden durch ein extern angelegtes elektrisches Bias-Feld beschleunigt. Der erzeugte Strom führt zu einem elektromagnetischen Impuls von 1 Picosekunde Dauer, ein Terahertz-Puls. Dieser Puls wandert via zwei Parabolspiegel (off-axis parabolic mirror) und Polarisatoren (Polarizer) zu dem Sample und wieder über zwei Parabolspiegel und einen Polarisator weiter zum Detektor. Der Detektor (Halbleiter) ermittelt den zeitlich veränderlichen THz Puls. Der Halbleiter ist so ausgelegt, dass er wie ein sehr schneller Schalter funktioniert. Der Detektor ist dann aktiv, wenn der fs-Puls und der THz Puls anliegen. Der dadurch resultierende Strom ist direkt proportional zum Terahertz-Feld. Eine charakteristische Eigenschaft jedes THz-TDS Systems ist, dass die Breite des erzeugten THz Puls (ca. 1 ps) viel länger ist als die Breite des prüfenden Laserpuls (<60 fs). Folglich wird jedes mal wenn der Laserpuls auf den Detektor trifft, nur ein Bruchteil des Terahertzpulses abgetastet. Um den ganzen Puls abtasten zu können, braucht man eine Verzögerung des Laserpulses. Diese Verzögerung wird durch die “Translation stage” erzeugt. Eine schnelle Fourier-Transformation des Zeitsignals liefert dann das Terahertzspektrum.

- **Laser:** Unser Laserstrahl hat eine Wellenlänge von 1560 nm, eine Wiederholrate von 100 MHz und einer Pulswidte von 50 fs.
- **Parabolspiegel:** Ein Parabolspiegel ist eine reflektierende Oberfläche, die dazu dient, ebene Wellen wie Licht, Schall oder Radiowellen, die parallel zur Achse einfallen, exakt in ihrem Brennpunkt zu sammeln. Da das Prinzip der Reflexion umkehrbar ist, lässt sich auch Strahlung von einer isotropen Punktquelle zu einem parallelen Strahl bündeln. Eine bekannte Anwendung von Parabolspiegel sind Satellitenschüssel [[Wikipedia contributors, 2022a](#)].
- **Hochpräzise Verzögerung:** Eine hochpräzise Verzögerung sorgt für die Zeitvariation, die wir für das Abtasten des THz-Puls brauchen. Sie besteht aus einem Retroreflektor (Tripelspiegel) in Kombination mit einem hochpräzisen Positionssensor auf einer Stage (Translation stage). Dieser Sensor zeichnet 50'000 Zeitstempel pro Sekunde mit einer Auflösung von 1.3 fs auf. Diese Zeitspempe werden mit den Daten von dem hochpräzisen Positionssensor synchronisiert.

3.4.2 THz-TDS mit Kunststoff

Polymere weisen im THz-Spektralbereich Polymer-spezifische Eigenschaften auf. Das liegt an dem mikroskopischen Aufbau der verschiedenen Kunststoffe. Dabei spielen zum Beispiel die Gerüstschwingungen (engl.: skeletal vibrations) eine Rolle. **Skeletal vibrations (SV)** sind rotatorische Schwingungen des Hauptkettensegments. Die spektrale Position und die Intensität des Schwingungsbandes hängt von der Flexibilität der Ketten und der nicht kovalenten Wechselwirkungen (Dipol-Dipol oder Wasserstoffbrücken) ab. **Wasserstoffbrücken** führen in der Regel zu einer Verschiebung der Peakfrequenz im mittleren Infrarot, die durch eine Streckung der kovalenten Bindungen entsteht. Allerdings können Wechselwirkungen mit Wasserstoffbrücken auch im THz Bereich beobachtet werden. Durch eine Rotation von permanenten Dipolen in einem elektrischen Feld entsteht die **Orientierungspolarisation (OP)**. Im Vergleich zu Flüssigkeiten ist die Oszillation der polaren Gruppen in den viskosen amorphen Domänen der Polymere jedoch aufgrund der kurzreichweiten Ordnung viel stärker eingeschränkt. Einerseits sind die Gruppen intramolekular durch kovalente Bindungen an oder in das Hauptkettensegment in ihrer Bewegung eingeschränkt. Andererseits werden die sich bewegenden Segmente durch benachbarte Makromoleküle in der gewundenen amorphen Phase sterisch¹ und elektrostatisch behindert. Daher wird diese Libration², die sich aus der behinderten Rotation der polaren Einheiten ergibt, als Flüssigkeitsgittermodus bezeichnet. Das breite Absorptionsband und das dispersive Verhalten in der Brechungsindexkurve ist beispielsweise auf die Libration der Estergruppe (PMMA) und der Phenylgruppe (PC, PS) zurückzuführen. PS weist kaum Dispersion und eine geringe, sehr breitbandige Absorption auf, da an der Phenyl-Seitengruppe nur ein kleines Dipolmoment vorhanden ist. Da diese Wechselwirkungen überwiegend aus den amorphen Domänen stammen, in denen die Makromoleküle zufällig verschränkt sind, tritt die Orientierungspolarisation auch als breites spektrales Merkmal auf [Peiponen et al., 2013]. Abbildung 2 zeigt den Extinktionskoeffizienten und den Brechungsindex einiger der gängigsten Kunststoffe.



©Modifiziert von [Wietzke et al., 2011]

ABBILDUNG 2 – Extinktionskoeffizient (κ) und Brechungsindex (n) verschiedener Polymere im Spektralbereich von 0,5 bis 3,5 THz. PVC (gelb markiert), PP (cyan) und PS (grün) sind hier zu finden.

¹Sterische Hinderung bezeichnet in der organischen Chemie den Einfluss der räumlichen Ausdehnung eines Moleküls auf den Verlauf einer Reaktion [Wikipedia contributors, 2022d].

²Libration ist eine Form der Hin-und-Herbewegung, auch als Taumelbewegung bezeichnet, bei der sich ein Objekt mit einer nahezu festen Ausrichtung immer wieder leicht hin und her dreht.[Wikipedia, 2019]

3.5 Maschinelles Lernen

Das Gebiet des maschinellen Lernens befasst sich mit der Frage, wie man Computerprogramme konstruieren kann, die sich mit der Erfahrung automatisch verbessern. Wir entscheiden uns hier für die folgende Definition von Lernen:

Ein Computerprogramm lernt aus Erfahrung E in Bezug auf eine Klasse von Aufgaben T und ein Leistungsmass P , wenn sich seine Leistung bei Aufgaben in T , gemessen durch P , mit der Erfahrung E verbessert [Mitchell, 1997].

Das maschinelle Lernen wird in der Regel als Teilbereich der künstlichen Intelligenz (KI) betrachtet (siehe Abbildung 3), bei dem es sich um eine Synthese von intelligentem menschlichem Verhalten in Maschinen handelt [Brown, 2021]. Je nach der sich häufig ändernden Definition von KI kann jedes System, das eine Aufgabe mit einem definierten Ziel ausführt, dabei flexibel auf sich ändernde Umgebungen reagiert und in der Lage ist, aus diesen Veränderungen zu lernen, als intelligent angesehen werden. Diese Definition beruht auf der Hypothese, dass logisches Denken reine Berechnung ist [Poole et al., 1998].

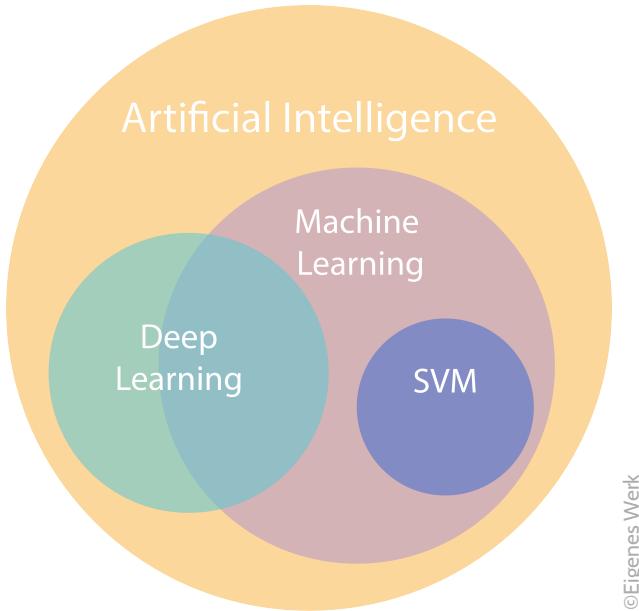


ABBILDUNG 3 – Die Abbildung zeigt den Bereich der künstlichen Intelligenz und einige seiner Teilbereiche.

3.5.1 Einführung in maschinelle Lerntechniken

Laut Bishop gibt es drei Hauptansätze für maschinelles Lernen, wenn es um das Training geht:

- *Supervised learning*: Anwendungen, bei denen die Eingänge mit den entsprechenden Ausgängen gepaart werden.
- *Unsupervised learning*: Anwendungen, bei denen den Eingaben keine entsprechenden Ausgaben gegenüberstehen.
- *Reinforcement learning*: Bei dieser Technik geht es darum, in einer gegebenen Situation geeignete Aktionen zu finden, um eine Belohnung zu maximieren.

Für jedes Problem gibt es die passenden Werkzeuge und Techniken, die in der Entwurfsphase des Systems entsprechend ausgewählt werden müssen [Bishop, 2006].

3.5.2 Training und Schätzung der Vorhersageleistung

Der Trainingsprozess spezialisiert den Algorithmus auf die Arbeit mit dem *Trainingsatz*, und seine Leistung bei Instanzen aus diesem Satz ist aufgrund von *Überanpassung* (Overfit) kein guter Indikator für die Vorhersageleistung bei ungesiehenen Daten. Wenn viele Daten zur Verfügung stehen, besteht ein Ansatz darin, einen *Validierungssatz* zu erstellen, um die Vorhersageleistung von Algorithmen zu vergleichen, die auf unterschiedlich komplexen Trainingsdaten trainiert wurden. Und selbst bei Verwendung eines Validierungssatzes kann es aufgrund der vielen Iterationen auf einem begrenzten Datensatz ratsam sein, einen *Testssatz* für die endgültige Bewertung abzusondern. Wenn die Datenmenge jedoch sehr begrenzt ist, ist die *Kreuzvalidierung* (Cross-validation) eine gute Möglichkeit, die Menge der Trainingsdaten zu maximieren und gleichzeitig eine stabile Schätzung der Vorhersageleistung zu erhalten, in dem dass verschiedene Teile der Daten zum Testen und Trainieren eines Modells in verschiedenen Iterationen verwendet werden. Auf diese Weise kann ein grosser Teil der Daten für das Training eingesetzt werden, während alle Daten für die Leistungsbewertung genutzt

werden [Bishop, 2006].

Eine wichtige Eigenschaft, die jeder gute Algorithmus für maschinelles Lernen haben muss, um zuverlässig zu sein, ist seine *Generalisierung*, d. h. seine Fähigkeit, Daten, die nicht im Trainingssatz enthalten sind, korrekt zu klassifizieren. Die meisten Algorithmen für maschinelles Lernen sind in der Regel recht gut darin, Hypothesen zu erstellen, die am Ende auswendig gelernt werden, d. h. sie klassifizieren die Instanzen in der Trainingsmenge korrekt, können aber die korrekten Ergebnisse in ungesehenen Daten nicht vorhersagen. Solche zu komplexen und spezifischen Systeme werden als Overfit bezeichnet [Cristianini and Shawe-Taylor, 2000]. Abbildung 4 ist ein typisches Beispiel für ein solches System. Hier stehen Blau und Rot für zwei verschiedene Klassen, die durch eine grüne überangepasste Linie und eine besser verallgemeinerte schwarze Linie getrennt sind. Im Einsatz wird die schwarze Linie neue Informationen besser trennen.

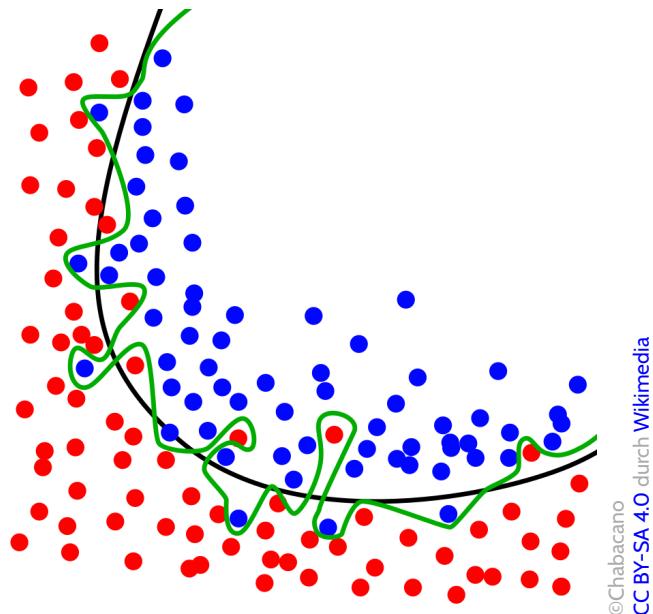


ABBILDUNG 4 – Ein Beispiel für das Overfitting der Trainingsdaten.

3.5.3 Merkmalsauswahl/Reduzierung der Dimensionalität

Daten aus der realen Welt sind komplex. Ereignisse, Objekte und Beobachtungen enthalten in der Regel viele Attribute, wodurch ein sehr komplexes Problem entsteht, das von einfachen Maschinen nicht gelöst werden kann. Aus diesem Grund ist die Merkmalsextraktion zur Dimensionalitätsreduzierung ein sehr wichtiger Vorverarbeitungsschritt bei der Arbeit mit Algorithmen für maschinelles Lernen. Unter Dimensionalität verstehen wir das Verhältniss zwischen Anzahl von Merkmale, z. B. Datenpunkte aus dem THz-Spektrum, und Anzahl von Beobachtungen oder Proben. Hochdimensionale Daten sind somit Beobachtungen, die viele Merkmale haben. Dabei handelt es sich im Wesentlichen um eine Auswahl von Informationen, die im Eingaberaum, d. h. den Rohdaten, enthalten sind und auf einen Merkmalsraum abgebildet werden sollen. Dies kann die Klassifizierungsaufgabe, also die Einteilung von Beobachtungen in zwei oder mehr Klassen in zweierlei Hinsicht erheblich vereinfachen: Verbesserung der Leistung und Vermeidung der Überanpassung [Cristianini and Shawe-Taylor, 2000].

3.5.4 Hyperebenen

Die Hyperebene ist eine Funktion, die zur Unterscheidung zwischen Merkmalen im Merkmalsraum verwendet wird. Unter der Annahme linear trennbarer Variablen und gegebener n Dimensionen kann die Hyperebene wie folgt definiert werden:

$$y = \sum_{i=1}^n w_i x_i + w_0 = \langle \mathbf{w}, \mathbf{x} \rangle + b \quad (1)$$

Dabei ist \mathbf{x} der Vektor, der die Eingaben enthält, also $\mathbf{x} = (x_1, \dots, x_n)$, und (\mathbf{w}, b) sind die Parameter, die die Funktion steuern. Diese Parameter sind diejenigen, die aus den Daten gelernt werden sollten. $\langle \mathbf{w}, \mathbf{x} \rangle$ ist das innere Produkt (auch Skalarprodukt genannt) der beiden Vektoren. Um diese Art von Klassifikator auf ein nichtlineares Problem anwenden zu können, wird das innere Produkt durch die gewünschte Kernel-Funktion ersetzt [Boser et al., 1992].

In Wirklichkeit definiert die Hyperebenenfunktion einen Unterraum in $n-1$, der im höherdimensionalen Raum schwierig zu visualisieren sein kann. Der Einfachheit halber werden wir uns auf einen linearen Klassifikator in \mathbb{R}^2 konzentrieren, somit wird die Hyperebene in $n - 1 \Rightarrow \mathbb{R}$ also eine Linie darstellen. Geometrisch würde diese Linie in \mathbb{R}^2 durch die Gleichung $\langle \mathbf{w}, \mathbf{x} \rangle + b = 0$ definiert werden.

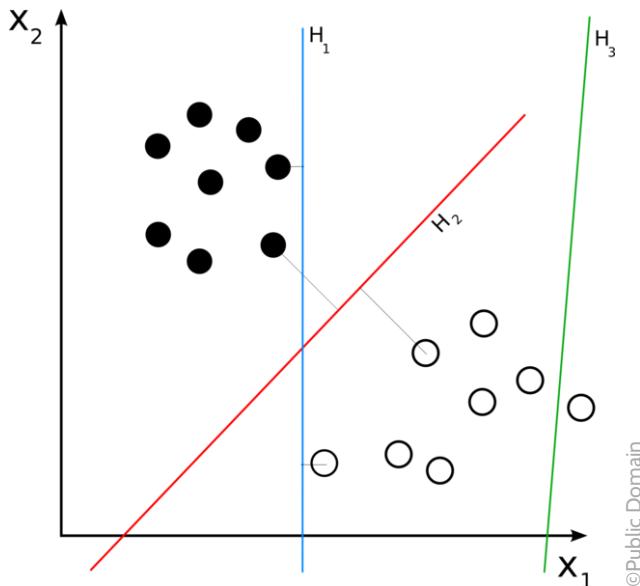


ABBILDUNG 5 – Hyperebenen in \mathbb{R}^2

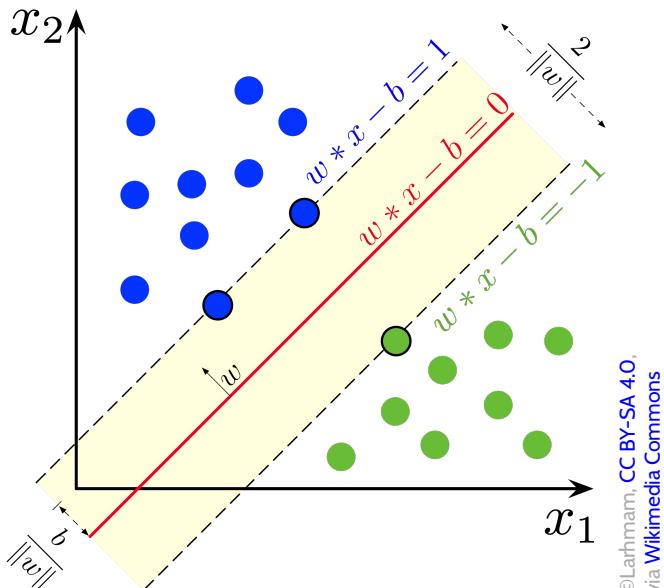
©Public Domain

Im Beispiel der Abbildung 5 trennen die drei Hyperebenen zwei Klassen auf eine schlechte Art ($H3$), eine akzeptable Art ($H1$) und eine gute Art ($H2$). Diese beste Hyperebene wird als Lösung für das Hard-Margin-Optimierungsproblem ermittelt. Um dies näher zu erläutern, beziehen wir uns auf die Abbildung 6. Die beste Hyperebene würde dann die beiden Klassen so unterteilen, dass der Raum links und rechts der Ebene zur nächsten Beobachtung so gross wie möglich wäre. Um dies zu erreichen, definieren wir zwei weitere Trennebenen in \mathbb{R}^2 :

$$\langle \mathbf{w}, \mathbf{x} \rangle + b = 1$$

$$\langle \mathbf{w}, \mathbf{x} \rangle + b = -1$$

Der Abstand zwischen diesen beiden Ebenen ist der Bereich, den wir maximieren wollen, und er ist geometrisch definiert als $\frac{2}{\|\mathbf{w}\|}$. Schlussendlich müssen wir also $\|\mathbf{w}\|$ so klein wie möglich wählen.



©Lathmam, CC BY-SA 4.0,
via Wikimedia Commons

ABBILDUNG 6 – Maximum-margin hyperebene für eine lineare 2-Klassen klassifizierung

In der Realität sind viele Probleme nicht linear trennbar. Aus diesem Grund benutzt man meistens den Soft-Margin-Optimierungsproblem. Im Gegensatz zum Hard-Margin-Optimierungsproblem erlaubt man hier den Hyperebenen, eine bestimmte Anzahl von Fehlern zu machen um den Spielraum so gross wie möglich zu halten, damit andere Punkte noch richtig klassifiziert werden können [Cristianini and Shawe-Taylor, 2000].

3.5.5 Kernel-basierte Methoden

Durch die Anwendung der Dimensionalitätsreduzierung kann man auch redundante Daten verdecken, was die Kategorisierung erschweren könnte. Aus diesem Grund ist es bei jedem Klassifizierungsproblem eine wichtige Herausforderung, ein ausgewogenes Verhältnis zwischen niedriger Dimensionalität zur Vermeidung von Überanpassung und der Beibehaltung wesentlicher Merkmale zu finden. Berechnungen in einem hochdimensionalen Raum, d. h. einem Raum mit vielen Merkmalen, die als einzelne Vektoren dargestellt werden, können sehr komplex sein und viel Zeit in Anspruch nehmen. Glücklicherweise gibt es einen mathematischen Trick, der heutzutage weit verbreitet ist: den Kernel-Trick. Mit diesem Trick wird eine implizite Repräsentation der Daten auf den Merkmalsraum übertragen, wodurch die mit der Auswertung der Merkmalskarte verbundenen Berechnungsprobleme umgangen werden. Der Kernel-Trick ist die Grundlage vieler Algorithmen des maschinellen Lernens, einschließlich Support Vector Machines (SVMs).

[Cristianini and Shawe-Taylor, 2000]

Einfach ausgedrückt, sind Kernel eine mathematische Technik, die es uns ermöglicht, unsere Lösungen im Merkmalsraum von linear auf nicht-linear zu erweitern. Dies wird in der Abbildung 7 mit einer Kernel Maschine \emptyset grafisch dargestellt. Einige Beispiele für Kernel-Funktionen sind Gauss, Linear, Polynom und Sigmoid.

In dieser Arbeit benutzen wir Gaußsche Kernels, die man durch die folgende Gleichung definieren kann:

$$K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right) \quad (2)$$

Dieser Kernel wird z. B. in der Gleichung 1 statt das innere Produkt beider Vektoren eingesetzt. Dabei ist σ ein Parameter; ein kleines Sigma führt zu einer scharfen Entscheidungsgrenze, während ein größeres Sigma zu einer weicheren Grenze führt [Schölkopf et al., 2004].

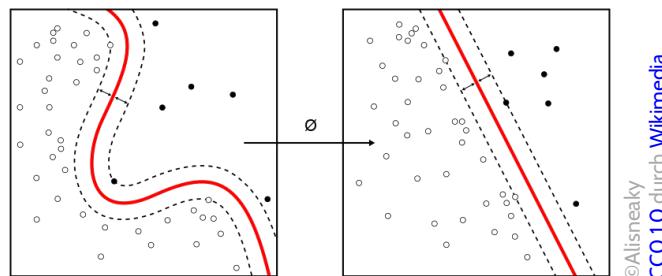


ABBILDUNG 7 – Kernel-Maschinen

3.6 Support Vector Machines (SVMs)

Support Vector Machines (SVMs) sind lernende Maschinen für Klassifizierungsprobleme. Die Eingaben werden auf einen sehr hochdimensionalen Merkmalsraum abgebildet, in dem eine Entscheidungsfläche durch die Anwendung von Hyperebenen, Kernels und Verallgemeinerungstechniken konstruiert wird.

Einige Vorteile von SVMs sind:

- Große Verallgemeinerungsfähigkeit bei hoher Dimensionalität.
- Es ist spechereffizient, da es Punkte im Merkmalsraum verwendet, die als Stützvektoren bezeichnet werden.
- Sehr vielseitig, da viele verschiedene Kernel-Funktionen verwendet werden können, um die Maschine an ein neues Problem anzupassen.

Insgesamt sind SVMs sehr leistungsfähige und universelle Lernmaschinen [Cortes and Vapnik, 1995].

Die gute Leistung der SVMs ist darauf zurückzuführen, dass sie im Gegensatz zu herkömmlichen Kernel-Methoden spärliche Kernel verwenden, die nur die relevantesten Punkte zur Trennung der Daten berücksichtigen und diese dann zur Erstellung von den Stützvektoren der Hyperebene verwenden [Bishop, 2006].

3.6.1 Mehrklassen-SVMs

Support Vector Machines sind grundsätzlich Zwei-Klassen-Klassifikatoren, d. h. sie können angeben, ob eine Instanz zu einer bestimmten Klasse gehört oder nicht (dann gehört sie durch Ausschluss zur anderen Klasse). Für die Lösung eines Problems mit mehr als zwei Ausgängen wurden verschiedene Ansätze entwickelt. Die gängigsten davon sind:

- One-versus-the-rest (Einer-gegen-die-anderen): Hier nimmt das Modell eine einzelne Klasse als positives Beispiel und den Rest als negatives Beispiel, iteriert dann durch die übrigen Klassen und wiederholt den Prozess, bis nur eine positive Klasse übrig bleibt.

- One-versus-one (Einer gegen einen): Hier werden viele SVMs trainiert, um jede Klasse gegen jede andere zu testen, wenn eine Eingabe gegeben wird. Die “Sieger”-Klasse wird dann als Ausgabe gegeben.

[Bishop, 2006]

Für unser Projekt haben wir Multiklassen-SVMs mit einem Gauss-Kernel verwendet (siehe Gleichung 2), in der One-versus-One-Konfiguration.

4 Arbeitsschritte

4.1 Probenbeschaffung

Zu Beginn unseres Projekts haben wir beschlossen, uns auf Plastikabfälle zu konzentrieren, die man normalerweise im Müll findet. Damit wollten wir eine mögliche Verzerrung der Stichproben³ in unseren Daten verringern. Gleichzeitig wollten wir das Potenzial dieser Methode zur Unterscheidung zwischen Polymeren mit ähnlichen physikalischen Eigenschaften wie Dichten aufzeigen.

Trainingsmenge

- **Polyethylenterephthalat ($N = 116$)⁴:**

PET war relativ leicht zu bekommen. Dieses Polymer wird von den Verbrauchern bereits gut von anderen Materialien getrennt. Allerdings werden nur Flaschen in Mülltonnen gesammelt. Diese haben eine sehr homogene Dickenverteilung. Um eine Überanpassung zu vermeiden, haben wir auch PET-Proben von Lebensmittelverpackungen und Schönheitsprodukten gesammelt. Wir haben nicht zwischen normalem PET und recyceltem PET (PETR/R-PET) unterschieden. Am Ende hatten wir $N = 105$ aus gesammelten Flaschen und $N = 11$ aus anderen Quellen.

- **Polypropylen ($N = 84$):**

PP wird häufig für Verpackungen verwendet. Wir haben eine Menge dieses Polymers gesammelt, indem wir den Müll in unseren Haushalten durchsuchten. Übliche Quellen waren Lebensmittelverpackungen sowie Plastikfolien, die in den Labors des Lehrbetriebs gesammelt wurden.

- **Polystyrol ($N = 44$):**

PS war etwas schwieriger zu finden. Es ist weithin in seiner expandierten Form (EPS) erhältlich, die aufgrund ihrer geringen Dichte leicht zu trennen ist. Molkereiprodukte wie Joghurt werden jedoch in der Regel in PS-Folien verpackt. Unser Datensatz besteht grösstenteils aus diesen weissen Folien ($N = 39$) sowie aus einigen transparenten PS-Stücke, die wir bei der Verpackung von Elektronikprodukten gefunden haben ($N = 5$).

- **Polyvinylchlorid ($N = 34$):**

PVC ist im Haushalt nicht so verbreitet wie PET oder PP. Dennoch wird es häufig für Rohre, Schwimmhilfen und Regenkleidung verwendet. Für die Messung haben wir mehreren farbigen PVC-Schaum Platten von Sheet Plastics bestellt.



ABBILDUNG 8 – PET (links oben), PP (rechts oben), PS (links unten) und PVC (rechts unten) Proben.

³Stichprobenverzerrung ist eine Verzerrung, bei der eine Stichprobe so erhoben wird, dass einige Mitglieder der Zielpopulation eine niedrigere oder höhere Stichprobenwahrscheinlichkeit haben als andere.

⁴ N ist die Anzahl der von jedem Polymer gesammelten Proben

Testmenge

Ein gut ausgewählter Testsatz ist ein Muss für die zuverlässige Validierung des Modells. Es ist darauf zu achten, dass der Satz heterogen ist, d. h. eine vielfältige Mischung unähnlicher Proben enthält. Auch wenn der Trainingssatz nicht symmetrisch war (er enthielt nicht die gleiche Anzahl von Proben für jede Kategorie), enthielt der Testsatz 5 Proben aus jeder Klasse. Keine Probe der Testmenge ist in der Trainingsmenge enthalten.

- **Polyethylenterephthalat ($N = 5$):**

Der PET-Testsatz enthielt 3 Proben von Plastikflaschen verschiedener Farben. Es enthielt grüne, braune und transparente Proben. Die beiden anderen Proben stammten von Lebensmittelverpackungen und waren transparent.

- **Polypropylen ($N = 5$):**

Für die PP-Testmenge wählten wir zwei vollständig transparente, eine undurchsichtige, eine grün gefärbte und eine schwarze Probe.

- **Polystyrol ($N = 5$):**

Der PS-Test bestand hauptsächlich aus weichen, weissen Proben und einer starren, transparenten Probe.

- **Polyvinylchlorid ($N = 5$):**

Die PVC-Testmenge enthielt alle verschiedenfarbigen Proben mit unterschiedlichen Dicken von 3 bis 5 mm und eine 10-mm-Probe. Eine Dicke, die für den Algorithmus in der Trainingsmenge unbekannt war.

4.2 Messungen und Auswertung

1. Zuerst haben wir die Proben von offensichtlichen Verunreinigungen befreit, indem wir sie mit einem trockenen Tuch abgewischt haben.
2. Danach haben wir unsere Proben auf eine handhabbare Grösse zugeschnitten, normalerweise Quadrate mit einer Seitenlänge von 20 bis 50 mm.
3. Dann haben wir unsere Proben gründlich mit optischen Wischtüchern und Ethanol gereinigt. Um etwaige Ethanolreste zu entfernen, tauchten wir die Proben kurz in destilliertes Wasser und wischten sie trocken.
4. Anschliessend haben wir die Proben mit unserem Transmissions-THz-TDS gemessen und die Daten in einer bezeichneten Datei gespeichert. Dies funktionierte bei allen Proben gut, da die meisten Polymere eine relativ hohe Transmission im THz-Spektrum aufweisen. Um ein aussagekräftiges Ergebnis zu erhalten, haben wir 500 Einzelmessungen gemittelt, die in etwa 15 Sekunden durchgeführt wurden.
5. Dann nahmen wir die Probe aus unserer TDS-Einrichtung und massen ihre Dicke mit einem Mikrometerschraube, die dann im Dateinamen jeder Probe vermerkt wurde.
6. Schliesslich wurde jede Probe mit einer eindeutigen Kennung versehen und in separaten Behältern für jede Polymer-Kategorie gelagert.

4.2.1 Extraktion von Brechungsindex und Absorption

Für die Extraktion des komplexen Brechungsindexes aus den Messdaten wurde ein etablierter Algorithmus verwendet, der in Python 3 implementiert ist [Mavrona, 2016]. Dieser Algorithmus basiert auf einer experimentellen Übertragungsfunktion, die anhand der Messung einer Probe in einem THz-TDS im Vergleich zu einer Referenzmessung, d. h. eine Messung ohne Probe im Strahlpfad, berechnet wird. Diese experimentelle Funktion wird dann mit einem theoretischen Modell der Ausbreitung des elektrischen THz-Feldes verglichen, um den komplexen Brechungsindex \tilde{n} zu ermitteln. Die Parameterextraktion beginnt mit der Berechnung einer experimentellen Übertragungsfunktion H_{exp} durch Division des Frequenzspektrums des Probencans E_s durch das Frequenzspektrum des Referenz-Scans E_{ref} (siehe Gleichung 3, wobei ω die entsprechende Frequenz ist). E_{ref} und E_s erhält man, indem man die in der Zeit gemessenen Signale mittels Fast Fourier Transformation (FFT) in den Frequenzbereich transformiert.

$$H_{exp}(\omega, \tilde{n}(\omega)) = \frac{E_s(\omega, \tilde{n}(\omega))}{E_{ref}(\omega, \tilde{n}(\omega))} \quad (3)$$

Der nächste Schritt besteht in der Berechnung der theoretischen Übertragungsfunktion (siehe Gleichung 4), die mit Hilfe der Newton-Raphson-Methode (ein Approximationsalgorithmus zur numerischen Lösung von nichtlinearen Gleichungen) an die experimentelle Funktion angepasst wird (d. h. $H_{exp} = H_{theo}$), um schliesslich den komplexen Brechungsindex der Probe \tilde{n}_s zu erhalten. Für die Berechnung von H_{theo} setzt man $\tilde{n}_{air} = 1$ für den Brechungsindex der Luft. L ist die Dicke der Probe und c die Lichtgeschwindigkeit.

$$H_{theo} = e^{-i(\tilde{n}_s - \tilde{n}_{air})\omega L/c} \frac{4\tilde{n}_s \tilde{n}_{air}}{(\tilde{n}_s + \tilde{n}_{air})^2} \quad (4)$$

Nachdem wir den Brechungsindex und die Absorption über das gesamte Spektrum (0,1-10 THz) erhalten haben, stellen wir das ursprüngliche Zeitsignal, seine Fourier-Transformation, den Brechungsindex und die Absorption für alle Proben mit Matplotlib (eine Python Library) dar. Wir tun dies, um nach gemessenen Ausreißern zu suchen, die auf eine falsche Handhabung während der Messung oder anderweitig verfälschte Daten hinweisen könnte. Diese fehlerhaften Proben wurden erneut gemessen, was zu plausiblen Ergebnissen für alle Proben führte. Die alten Daten wurden verworfen und durch die neuen Messungen ersetzt.

Nach der Überprüfung wurden der Brechungsindex und die Absorption für den ganzen Frequenzbereich in einer eigenen Datei für jede Probe gespeichert.

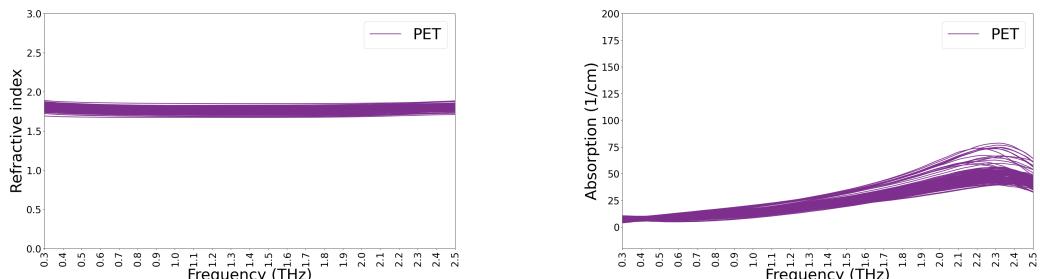


ABBILDUNG 9 – Brechungsindex (links) und Absorption (rechts) von PET

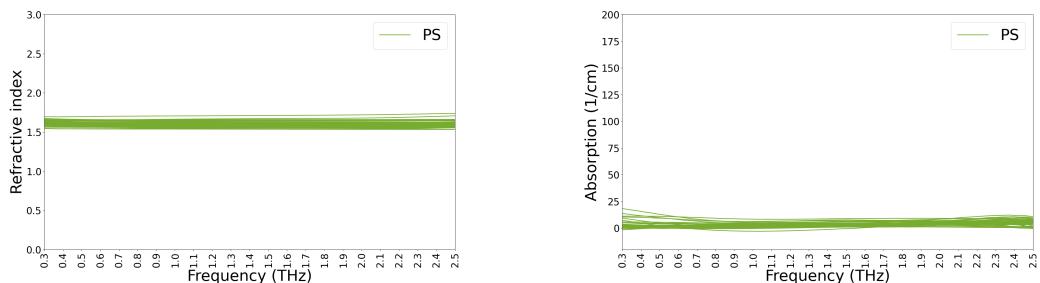


ABBILDUNG 10 – Brechungsindex (links) und Absorption (rechts) von PS

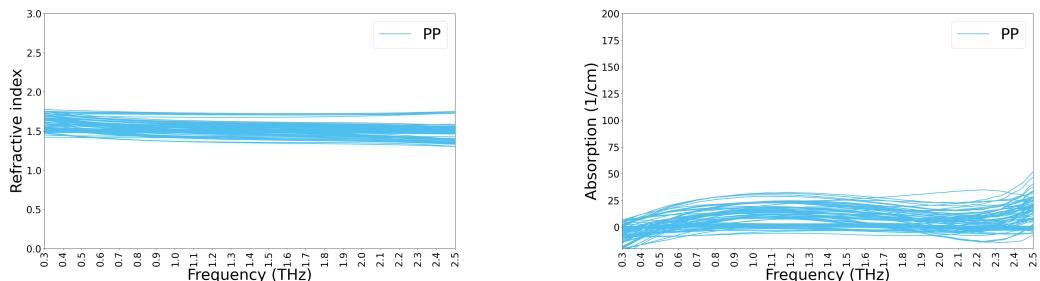


ABBILDUNG 11 – Brechungsindex (links) und Absorption (rechts) von PP

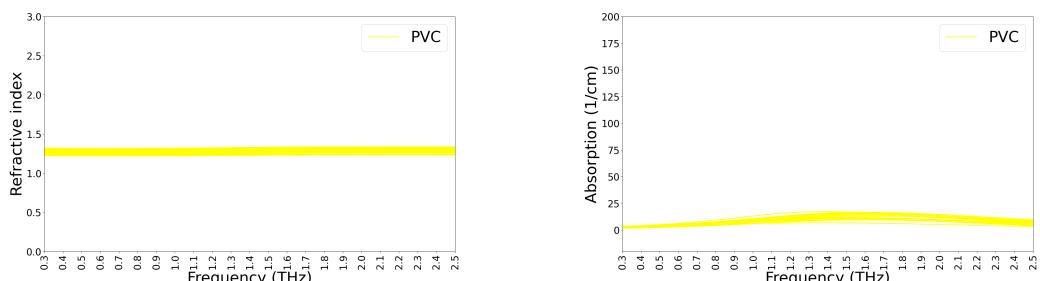


ABBILDUNG 12 – Brechungsindex (links) und Absorption (rechts) von PVC

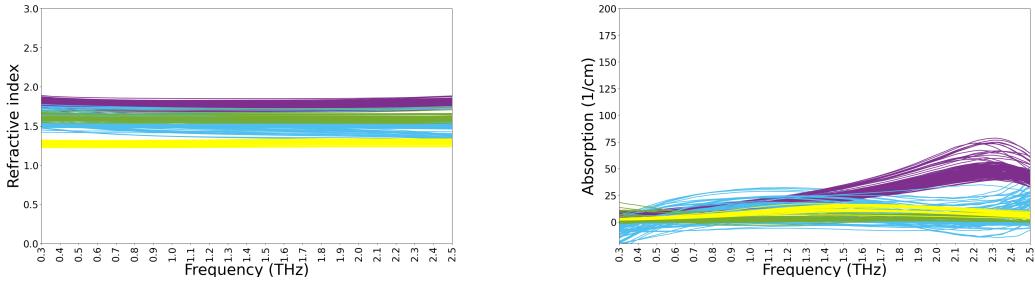


ABBILDUNG 13 – Brechungsindex (links) und Absorption (rechts) von PET (lila), PVC (gelb), PP (cyan) und PS (grün).

4.2.2 Merkmalsextraktion mit Python

Nachdem wir den gesamten Datensatz mit allen verschiedenen Kategorien von Polymeren visualisiert haben, wurden eindeutige Merkmale erkennbar. Wir haben die charakteristischsten von ihnen ausgewählt, um die Vorhersageleistung und die Generalisierung im maschinellen Lernmodell zu optimieren. Die für das maschinelle Lernen extrahierten Merkmale waren:

- Brechungsindex bei 2,3 THz:
Bei der Bestimmung des Brechungsindexes bei 2,3 THz haben wir das Ende des gemessenen Spektralbereichs berücksichtigt. Dabei verwenden wir den Spektralbereich bis 3 THz, da die Absorption des Wassers aus der Luftfeuchtigkeit die Messung bei höheren Frequenzen verzerrt.
- Absorption bei 2,3 THz:
Dieses Merkmal wurde wegen der hohen Absorption von PET bei dieser Frequenz ausgewählt.
- Standardabweichung der Absorption von 0,5 bis 2,5 THz:
PET und PP, im Gegensatz zu PS, weichen in diesem Bereich deutlich vom Durchschnitt ab.
- Standardabweichung des Brechungsindexes von 0,5 bis 2,5 THz:
PS hat in diesem Spektralbereich eine sehr geringe Standardabweichung von seinem Brechungsindex.
- Mittlere Absorption von 0,5 bis 1,5 THz:
Der Durchschnitt der Absorption von PVC ist in diesem Spektralbereich höher als bei PS.
- Absorption bei 1,3 THz:
PP hat eine etwas höhere Absorption bei 1,3 THz, die dann bei 2,3 THz wieder abnimmt.

Diese Merkmale schnitten bei der Kategorisierung dieser 4 Polymere in unserem Algorithmus von allen ausprobierten Eigenschaften am besten ab. Das bedeutet nicht, dass es nicht noch bessere Merkmale gibt (siehe Abschnitt 6.4).

4.2.3 Anwendung von SVMs

Für die Modellerstellung haben wir den Classification Learner von MATLAB R2020b verwendet. Mit dieser Anwendung können wir schnell unser Trainingsset hochladen und aus einer Vielzahl von Modellen für maschinelles Lernen auswählen, die trainiert werden sollen. Die Benutzeroberfläche ist intuitiv und die Berechnungszeit ist relativ kurz. Wir haben uns für MATLAB entschieden, weil es einfach zu bedienen ist und es uns ermöglicht, schnell viele verschiedene Modelle zu testen. Dies hat den Vorteil, dass man seine Daten vielen Algorithmen aussetzen kann, die möglicherweise etwas besser mit den gewählten Merkmalen arbeiten. Außerdem stand uns eine MATLAB-Lizenz über den Software-Shop unseres Lehrbetriebs zur Verfügung.

Die Modellerstellung in MATLAB beginnt mit den Parameterdefinitionen für die Funktion templateSVM():

```
template = templateSVM(...  
'KernelFunction', 'gaussian', ...  
'PolynomialOrder', [], ...  
'KernelScale', 2.4, ...  
'BoxConstraint', 1, ...  
'Standardize', true, ...  
'SaveSupportVectors', true);
```

- **KernelFunction:** Legt die Funktion fest, die das Programm verwendet, um die explizite Abbildung zu vermeiden, die erforderlich ist, damit lineare Lernalgorithmen eine nichtlineare Entscheidungsgrenze lernen. In diesem Fall wird *Gaussian* ausgewählt, sodass Gleichung 2 statt das innere Produkt im Merkmalsraum verwendet wird.
- **PolynomialOrder:** Explizit zu [] gesetzt, da den Kernel *Gaussian* ist.
- **KernelScale:** Die Software teilt alle Elemente der Prädiktormatrix X durch den Wert von KernelScale. Mit dem Wert 2.4 bestimmen wir einen "medium Gaussian".
- **BoxConstraint:** Ist ein Parameter, der die maximale Strafe für marginverletzende Beobachtungen (siehe Abbildung 6) steuert, was dazu beiträgt, eine Überanpassung (Abbildung 4) zu verhindern.
- **Standardize:** Die Software zentriert und skaliert jede Spalte der Vorhersagedaten (X) mit dem gewichteten Spaltenmittelwert bzw. der Standardabweichung

[Support vector machine template, 2022]

Nachdem diese Parameter festgelegt wurden, muss das Modell mit dem ECOC-Mehrklassenmodus (error-correcting output codes) angepasst werden. Dies geschieht über die Funktion fitcecoc(), die die soeben erstellte Vorlage, unsere Merkmalstabelle, die zu klassifizierenden Kategorien und einige weitere Parameter aufnimmt.

```
classificationSVM = fitcecoc(...  
predictors, ...  
response, ...  
'Learners', template, ...  
'Coding', 'onevsone', ...  
'ClassNames', categorical({'PET'; 'PP'; 'PS'; 'PVC'}));
```

- **predictors:** Eine Liste mit den Merkmalen, die wir im Modell nutzen wollen. Hier:

```
predictorNames = {'nAt2_3THz', 'alphaAt2_3THz', ...  
'alphaStdDevfrom0_5to2_5THz',  
'nStdDevfrom0_5to2_5THz', ... 'alphaAvgfrom0_5to1_5THz',  
'alphaAt1_3THz'};  
predictors = inputTable(:, predictorNames);
```

- **response:** Hier geben wir die gewünschte Ausgabe unseres Modells ein, d. h. die Kategorien, die wir am Ende erhalten möchten.

```
response = inputTable.Category;
```

- **Learners:** Wir legen fest, welches Modell wir trainieren möchten; dies ist der *returned object* unserer templateSVM().
- **Coding:** Hier weisen wir unser Modell an, mit einem One-Vs-One-Ansatz für die Multiklassen-Klassifikation vorzugehen.
- **ClassNames:** Die Bezeichnungen der einzelnen Klassen

[fitcecoc, 2022]

4.3 Agile Robots for Tomorrow's Lab (ARTLab)

Die Dicke jeder Probe musste einzeln gemessen werden, um diese Informationen in den Algorithmus zur Bestimmung der Absorption und des Brechungsexponenten einzugeben. Dazu verwendeten wir eine Mikrometerschraube an der gleichen Stelle der Probe, die wir im TDS gemessen haben. Dann wurde die Dicke in der Namensdatei mit den Spektralinformationen gespeichert. Dieser Vorgang war sehr zeitaufwendig, aber wir konnten ihn gut in unsere Messungen integrieren, da wir ihn durchführten, während wir gleichzeitig die nächste Probe im TDS massen. Dennoch wollten wir diesen Prozess zumindest teilweise automatisieren, um die Anpassung an einen größeren Massstab zu erleichtern. Wir fanden dies ein gutes Beispiel, um es in unser ARTLab-Projekt an der Empa zu integrieren.

Das Ziel von ARTLab ist es, eine agile Roboterplattform zu entwickeln, um die Schnelligkeit und Reproduzierbarkeit von sich wiederholenden Laborarbeiten zu verbessern und diese in die Ausbildung von Lernenden zu integrieren. Der Zugang zu einem sechsachsigen Industrieroboter mit hoher Wiederholgenauigkeit ($\pm 5 \mu\text{m}$) [Mecademic, 2022] ermöglichte es uns, ein automatisches System zu entwickeln, das die Dicke unserer Proben schnell und zuverlässig messen konnte. Zu diesem Zweck haben wir unseren Meca500-Roboterarm mit einem inkrementalen Messtaster ST 3078 von Heidenhain (Längemessgerät) ausgestattet, das eine Genauigkeit von $\pm 1 \mu\text{m}$ hat [Heidenhain AG, 2022]. Der Messaufbau besteht aus einer Platte, auf die die Probe gelegt wird, und dem Meca500, der das ST 3078 hält, wie in Abbildung 14 dargestellt. Nach dem Auflegen der Probe bewegt der Roboter den Messtaster linear nach unten, und die Dicke der Probe wird dann in Bezug auf den Nullpunkt auf der Oberfläche der Messplatte gemessen.

Um diesen Aufbau zu testen, haben wir 100 aufeinanderfolgende Messungen an einer Probe mit einer Dicke von 3,150 mm vorgenommen, die wir mit einer Mikrometerschraube gemessen haben, und ihre Standardabweichung ($\sigma = 1,04 \mu\text{m}$) berechnet. Die Ergebnisse sind in dem Histogramm in Abbildung 15 dargestellt.

Der Roboterarm könnte auch dazu verwendet werden, die Messung weiter zu automatisieren, z. B. durch Positionierung der Probe im TDS.

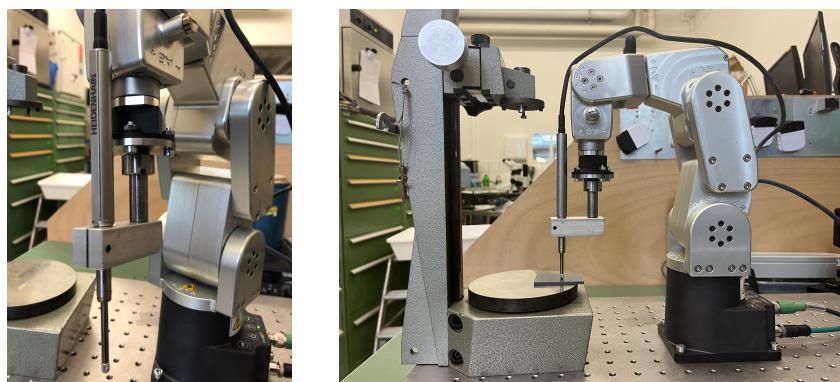


ABBILDUNG 14 – Einsatz vom Meca500 und Heidenhain ST3078 bei der Dickenmessung

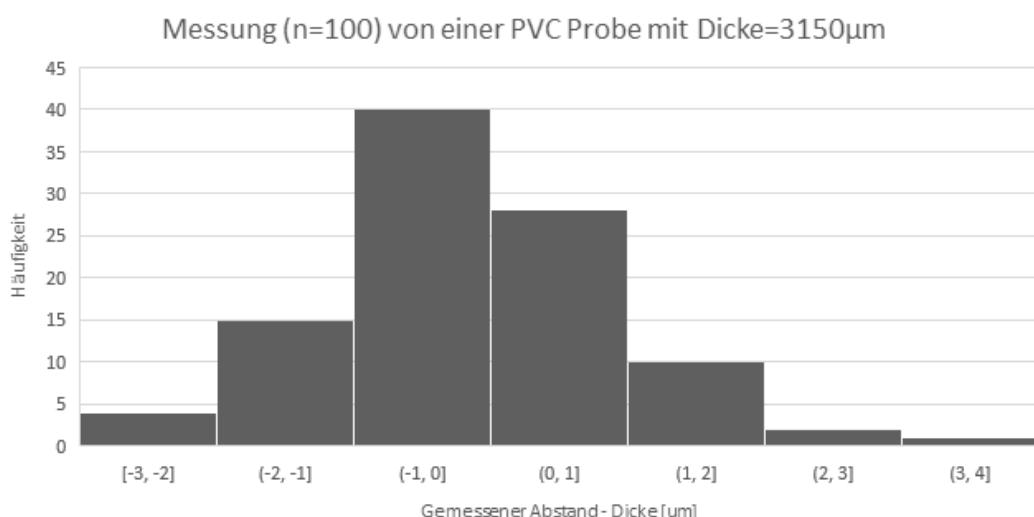


ABBILDUNG 15 – Histogramm einer Messung mit dem automatischen System

5 Ergebnisse

Die Validierungsgenauigkeit unseres endgültigen Modells auf der Trainingsmenge ($N = 278$) betrug 98.56%. In der Testmenge ($N = 20$) war das Modell in der Lage, die entsprechende Klasse jeder Probe genau vorherzusagen, also mit einer Genauigkeit von 100%. Die Abbildung 16 zeigt zwei Ansichten vom Merkmalsraum in 3D. Hier wird deutlich, dass die Klasse PET (lila) leicht von den anderen 3 Klassen getrennt werden konnte. PVC (gelb) konnte ebenfalls leicht von allem anderen getrennt werden. PP (cyan) und PS (grün) waren schwieriger, da sie sich oft überschneiden. Ein Teil des

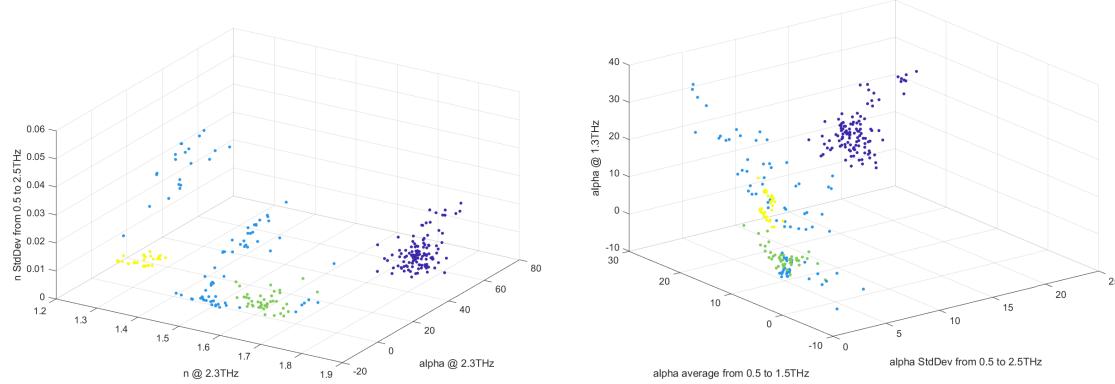


ABBILDUNG 16 – Zwei niedrigdimensionale Ansichten des 6-dimenionalen Merkmalsraums

Merkmalsraums wird leichter erkennbar, wenn es in einen zweidimensionalen Raum übertragen wird, wie in der Abbildung 17 dargestellt. Aufgrund der Ähnlichkeit der gewählten Merkmale von PS und PP im Trainingssatz hatte das Modell

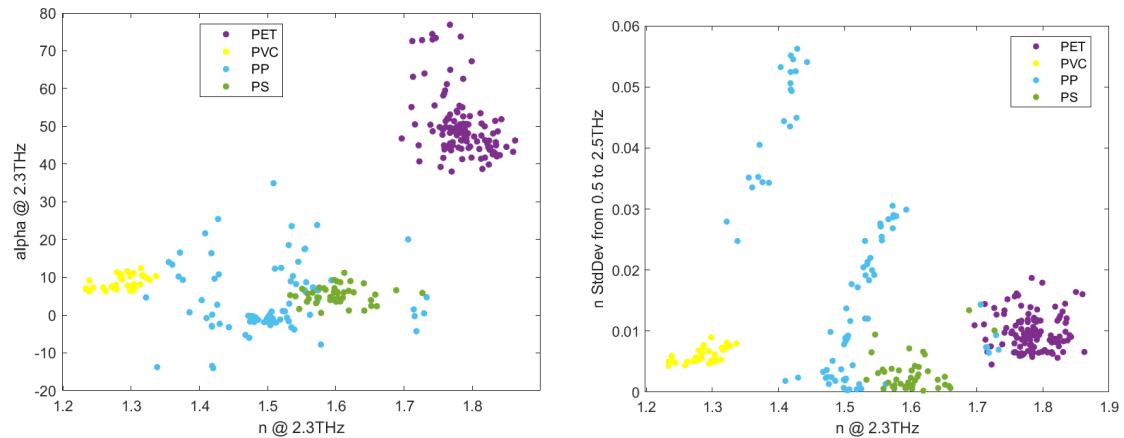


ABBILDUNG 17 – Zwei der zweidimensionalen Ansichten des Merkmalsraums

Schwierigkeiten, zwischen diesen beiden zu unterscheiden. Dies wird in der Konfusionsmatrix des Modells deutlich, die in Abbildung 18 dargestellt ist. In einer Konfusionsmatrix wird die vorhergesagte Klasse gegen die wahre Klasse aufgetragen. Auf diese Weise lässt sich leicht erkennen, welche Kategorien mit dem aktuellen Modell schwer vorherzusagen sein könnten. Die Konfusionsmatrix des Trainingsatzes zeigt, wie das Modell durch die Vorhersage einer PS-Probe anstelle der wahren PP-Probe verwirrt wurde.

Die für die Vorhersage der Testmenge benötigte Zeit wurde mit den tic/toc-Funktionen in MATLAB gemessen. Für die 20 Testmuster auf dem verwendeten System⁵ benötigte das Modell 14 ms, um die Menge zu kategorisieren. Dies entspricht etwa 0.7 ms pro Probe.

⁵HP EliteBook 840 G5, Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz, 1992 Mhz, 4 Core(s), 8 Logical Processor(s), 16GB RAM, Microsoft Windows 10 Enterprise

Confusion Matrix - Training Set

		PET	PP	PS	PVC
True Class	PET	116			
	PP		83	1	
	PS			44	
	PVC				34

Confusion Matrix - Test Set

		PET	PP	PS	PVC
True Class	PET	5			
	PP		5		
	PS			5	
	PVC				5

ABBILDUNG 18 – Konfusionsmatrizen von der Trainingsmenge (links) und Testmenge (rechts)

6 Diskussion

6.1 Bestimmung der Probendicke

Wir haben alle unsere Proben mit dem Transmissionsaufbau unseres THz-TDS gemessen. Auf diese Weise sammeln wir die Informationen des Pulses, der den gesamten Weg durch die Probe zurückgelegt hat, da die meisten Polymere eine hohe Transmission und einen relativ geringen Reflexionsgrad im THz-Spektrum aufweisen. Dies erschwert jedoch den Einsatz dieser Technologie, da die Transmissionsmessung im grossen Massstab schwieriger anzuwenden ist. Ausserdem mussten wir bei unserem Algorithmus die Dicke jeder Probe separat messen, was wir in unserem Projekt zwar manuell taten, was aber im kleinen Massstab durch den Einsatz von Robotern im Labor leicht automatisiert werden könnte.

Eine mögliche Lösung für die Dickenmessung wäre die Verwendung eines Algorithmus, der die Dicke der Probe mit Hilfe der Mehrfachreflexionen in einem iterativen Ansatz berechnen kann, wie in [Scheller et al., 2009]. Es ist wichtig zu wissen, dass die Informationen, die zur Schätzung der Probendicke benötigt werden, bereits in den Messdaten enthalten sind. Die resultierende Kurve enthält die Mehrfachreflexionen im Inneren des Materials, die jedoch schwer direkt zu ermitteln sind. Zusammen mit unserem Experten Gregory Gäumann vom nationalen Wettbewerb Schweizer Jugend Forscht haben wir die Möglichkeit untersucht, eine kommerzielle Software für die Abschätzung der Dicke zu verwenden. Die Software heisst RefFIT [Kuzmenko, 2005] und ist für das Fitting optischer Spektren aus Reflexions-, Transmissions- und Ellipsometriemessungen konzipiert. All dies kann über die grafische Benutzeroberfläche und unter Verwendung speziell geschriebener Makros zur Anpassung der Spektren erfolgen. Die Dicke wird durch iterative Anpassung der gemessenen Spektren an eine theoretische Kurve ermittelt, die anhand der Permittivität des Kunststoffs berechnet wird. Bei der Untersuchung dieser Methode haben wir Daten zur Permittivität von unseren eigenen Proben und Werte aus der Literatur verwendet. Das Modell ist dann in der Lage, die Dicke der Probe für eine bestimmte Art von Polymer vorherzusagen. Um sicherzustellen, dass die richtige Permittivität, und somit der richtige Kunststoff für die Anpassung des Spektrums verwendet wird, wird die Permittivität des Kunststoffes mit dem kleinsten Chi-Quadrat-Wert verwendet, einem statistischen Mass für den Unterschied zwischen den beiden Kurven. Wie in der Abbildung 19 zu sehen ist, stimmt das von RefFIT berechnete Modell sehr gut mit dem gemessenen Spektrum überein. Die von RefFIT gefititte Dicke stimmt sehr gut mit der manuell gemessenen Dicke überein und weist eine Abweichung von nur $4 \mu\text{m}$ auf. Weitere Untersuchungen zur Bestimmung der Dicke mittels Time-Domain Fitting müssten gemacht werden um den Einfluss variabler Einfallswinkel auf das Ergebnis zu untersuchen.

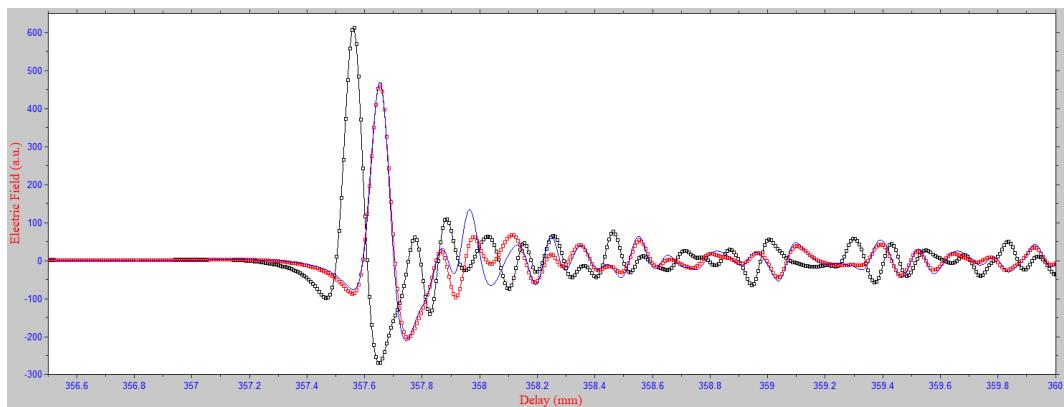


ABBILDUNG 19 – RefFIT zeigt ein Fenster mit der Kurvenanpassung. Die Referenzmessung ist in schwarz und die Polymermessung in rot dargestellt. Das durch RefFIT berechnete Modell mit gefitteter Dicke wird in blau dargestellt.

6.2 Reflexionsmessung

Ein weiteres Problem bei der Transmissionsmessung besteht darin, dass sich die Probe zwischen dem Sender und dem Sensor befinden muss. Dies erschwert den Einsatz dieser Methode in grossem Massstab, da ein komplexeres optisches oder mechanisches System erforderlich wäre, um die Proben für die Messung zu positionieren. Eine Möglichkeit, dieses Problem zu lösen, wäre, stattdessen mit einem reflektierenden System zu arbeiten. Dies ermöglicht es, dass sich Sender und Empfänger auf derselben Seite der Probe befinden, was eine schnellere und einfache Messung ermöglicht. Es gibt jedoch eine Reihe von Nachteilen, wenn man mit Reflexion statt mit Transmission arbeitet:

- Polymere haben in der Regel eine sehr hohe Transmission und eine relativ geringe Absorption [Peiponen et al., 2013]. Dadurch ist aber die Reflexion auch klein, da $\alpha + \rho + \tau = 1$ (siehe Abbildung 20). Somit würde unser Signal schwächer sein als bei der Transmissionsmessung.
- Weil in Reflexion nur sehr wenig THz-Strahlung absorbiert werden kann, und das Signal gemäss dem ersten Punkt generell auch schwächer wäre, würde es schwierig werden zwischen den Kunststoffen unterscheiden zu können.

Theoretisch wären Reflexionsmessungen im grossen Massstab einfacher zu realisieren. Um dieses System erfolgreich anwenden zu können, sind weitere Forschungsarbeiten erforderlich. Als Grundlage könnte beispielsweise die Messung der Dicke verschiedener Lack- und Farbschichten in der Autoindustrie dienen [Krimi et al., 2016].

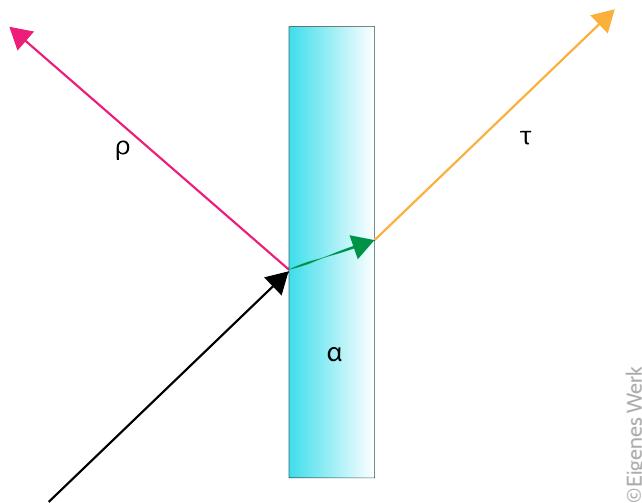


ABBILDUNG 20 – Reflexion ρ , Transmission τ und Absorption α dargestellt

6.3 Verzerrung der Stichprobe und Verallgemeinerungsfähigkeit

Eine Stichprobenverzerrung (sampling bias) ist eine Verzerrung, bei der eine Stichprobe so erhoben wird, dass einige Mitglieder der vorliegenden Zielpopulation eine niedrigere oder höhere Stichprobenwahrscheinlichkeit haben als andere. Stichprobenverzerrungen entstehen durch die Art und Weise, wie wir unsere Stichproben erheben [Lane, 2003]. Ein Beispiel für diese Verzerrung findet sich in Abbildung 21, wo die Zielpopulation (in unserem Fall Polymere) möglicherweise

nicht homogen verteilt ist und wir aufgrund unserer Methode der Probenahme aus dieser Population eine Probenmenge erhalten, die nur eine begrenzte Anzahl der vorhandenen Merkmale aufweist. Dies bedeutet, dass unser Trainingssatz möglicherweise nicht für alle Variationen der im Abfall vorkommenden Polymermerkmale repräsentativ ist.

Im folgenden gehen wir auf die beiden möglichen Stichprobenverzerrungen ein, die in unserem Fall von Bedeutung

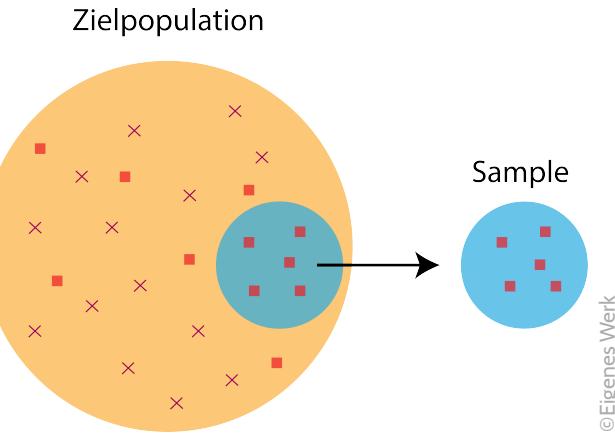


ABBILDUNG 21 – Stichprobe, die nicht repräsentativ für die Merkmale der Zielpopulation ist.

sein könnten:

1. **Zusatzstoffgehalt:** Dies sind chemische Verbindungen, die zur Verbesserung der Funktionalität und der Alterungseigenschaften des Polymers hinzugefügt werden. Die gebräuchlichsten Additive sind: Weichmacher, Flammeschutzmittel, Antioxidantien, Säurefänger, Licht- und Hitzestabilisatoren, Schmiermittel, Pigmente, Antistatika, Gleitmittel und Wärmestabilisatoren [Hahladakis et al., 2018]. Es ist wichtig zu bedenken, dass viele von diesen verschiedenen Chemikalien die THz-Spektraleigenschaften der gemessenen Polymere verändern können [Peiponen et al., 2013]. Es ist möglich, dass das ML-Modell durch die ausschliessliche Verwendung von Kunststoffabfällen aus der Schweiz, insbesondere aus dem Kanton Zürich (ausser PVC), dazu neigt, nur die spezifischen Polymere mit den in dieser Region gefundenen Additiven richtig zu kategorisieren.
2. **Wassergehalt des Polymers:** Polare Flüssigkeiten wie Wasser absorbieren THz-Strahlung stark [Kindt and Schmuttenmaer, 1996]. Das bedeutet, dass unsere Messungen je nach dem Wassergehalt des gemessenen Polymers stark variieren können. Unser Trainingsset ist möglicherweise nicht repräsentativ für die verschiedenen Wassergehalte, die Kunststoffabfälle aufweisen können, zum Beispiel in einer Umgebung mit hoher Luftfeuchtigkeit. Es ist auch möglich, dass wir unseren Probensatz verzerrt haben, indem wir die Proben vor der Messung mit destilliertem Wasser gereinigt haben, wodurch sich ihr Wassergehalt erhöht und unsere spektralen Eigenschaften verändert haben könnte.

Der beste Weg, diese Verzerrung zu überwinden, wäre eine grössere Trainingsmenge, die möglicherweise viel mehr verschiedene Merkmale enthält, die in denselben Polymeren vorkommen. Dies kann durch die Messung einer grösseren Anzahl von Proben erreicht werden, die an verschiedenen Orten und von verschiedenen Herstellern, zu verschiedenen Jahreszeiten und in verschiedenen Zuständen entnommen wurden.

Eine andere Möglichkeit wäre die Verwendung von Korrekturtechniken in unserem ML-Algorithmus, wie sie von [Cortes et al., 2008] vorgeschlagen wurden. Dies würde jedoch das Problem der verzerrten Stichprobe nicht gänzlich lösen.

Dies berührt auch die Problematik der Generalisierungsfähigkeit unserer SVMs. Wie in Abschnitt 3.5.2 erläutert, kann es zu einer Überanpassung kommen, wenn der Algorithmus sich zu sehr auf die Trainingsmenge spezialisiert und in der Testmenge schlecht funktioniert. Eine Möglichkeit, dieses Problem zu vermeiden, ist eine gute Wahl des Kernels und seiner Hyperparameter. In unserem Fall hat sich der Gauss-Kernel mit einem MATLAB-Parameter "KernelScale" von 2.4 für unseren Probensatz am besten erwiesen.

Es ist jedoch zu beachten, dass diese Faktoren hauptsächlich unsere Einteilung in die breitesten Kategorien, d. h. PP, PVC, PS und PET, einschränken. Der Algorithmus versucht, jede neue Probe einer dieser Kategorien zuzuordnen, auch wenn es sich um ein völlig anderes Polymer handelt. Diese Merkmale können aber auch zur weiteren Unterscheidung zwischen den Polymertypen verwendet werden. Man kann sich die Klassifizierung einzelner Polymere mit einem bestimmten Zusatzstoff vorstellen, was den Recyclingprozess durch die Vermeidung unerwünschter Verunreinigungen weiter vereinfachen könnte.

6.4 Informationsverlust durch Merkmalsauswahl

Wie in Abschnitt 3.5.3 erläutert, ist die Extraktion relevanter Merkmale ein wichtiger Schritt, der das Problem für Algorithmen des maschinellen Lernens vereinfachen kann, wenn sie mit einer Klassifizierungsaufgabe konfrontiert werden. Bei der Auswahl der Merkmale, die wir zur Klassifizierung der verschiedenen Polymere verwenden wollen, übersehen wir jedoch viele Informationen, die anfangs nicht sehr relevant erscheinen, aber dennoch nützliche Merkmale zur Unterscheidung der Klassen enthalten könnten. Dies könnte dadurch gelöst werden, dass die Gesamtheit der erhaltenen Spektraldaten aus Tausenden von Datenpunkten besteht, die eine bestimmte Probe charakterisieren. All diese Informationen erhöhen jedoch die für den Lernprozess des maschinellen Lernalgorithmus erforderliche Rechenleistung erheblich. Aufgrund der limitierten Rechenleistung unseres Computers konnten wir diese Option nicht erfolgreich testen.

Dennoch gibt es Algorithmen der künstlichen Intelligenz, die in der Lage sind, die komplexen Merkmale zu verarbeiten, die sich bei der Analyse des gesamten Datensatzes der Spektralanalyse ergeben. Ein möglicher Kandidat für diese Art von Lernaufgabe wäre ein künstliches neuronales Netz (ANN), das komplexe Merkmale klassifizieren kann und in der Klassifizierung von Patterns (z. B. in der Bilderkennung) weit verbreitet ist [Kim, 2010].

Ursprünglich wollten wir ein Convolutional Neural Network (CNN) verwenden, um die spektralen Merkmale wie Bilder zu analysieren, d. h. je ein 2D-Array für die Absorptions- und Brechungsindexspektren jeder Probe. Dieser Ansatz scheiterte an unserer relativ kleinen Trainingsmenge, da CNNs für den realen Betrieb eine grosse Menge an Proben benötigen.

Eine mögliche Erweiterung dieses Projekts wäre die Verwendung unserer SVMs, um viele weitere Proben zu kategorisieren und zu beschriften, wie es in einer industriellen Anwendung der Fall wäre, und dann die beschrifteten Spektralinformationen zum Trainieren eines ANN im Hintergrund zu verwenden. Mit der Zeit würden sich so genügend Informationen ansammeln, um das ANN nutzbar zu machen und Kunststoffe anhand dieser ungewöhnlichen Merkmale besser zu trennen. Ein solcher Ablauf ist in Abbildung 22 dargestellt.

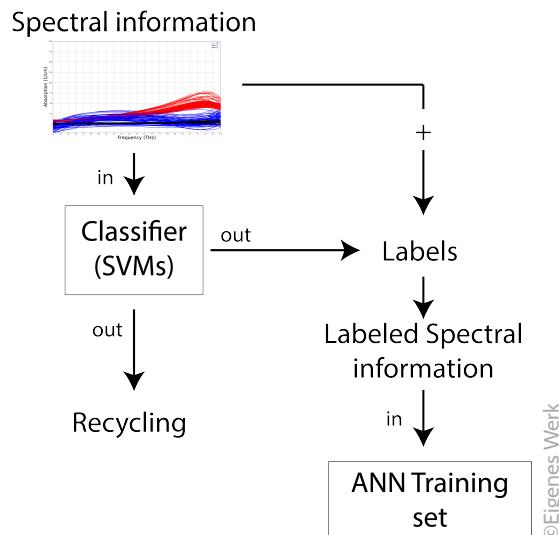


ABBILDUNG 22 – Prozessablauf mit einer Feedback-Schleife für das Training eines ANN

6.5 Exploration in der industriellen Anwendung

Wie bereits erwähnt, müssen neue und effiziente Methoden zur Klassifizierung von Polymeren entwickelt werden, um sie zu recyceln. Im Rahmen unserer weiteren Arbeit, die wir mit Hilfe unseres Experten von Schweizer Jugend Forscht durchgeführt haben, haben wir beschlossen, uns an ein Unternehmen zu wenden, das sich auf das Recycling von Kunststoffen spezialisiert hat. Wir hatten die Möglichkeit Patrik Ettlin, Leiter Kommunikation, der InnoRecyclingAG einige Fragen bezüglich der Kunststofftrennung zu stellen. Die InnoRecycling AG ist eines der führenden Unternehmen in der Schweiz im Bereich Kunststoffentsorgung und Recycling. Ihr Materialkatalog umfasst über 120 unterschiedliche Kunststoffsorten und -qualitäten. Herr Ettlin war sehr hilfsbereit und beantwortete einige der Fragen, die für unser Projekt wichtig sind.

Welche Verfahren verwenden Sie für die Kunststofftrennung?

In unserer Sortieranlage in Lustenau wird der in der Schweiz gesammelte Haushalt-Kunststoff in insgesamt 14 verschiedene Kunststoff-Fraktionen getrennt und den verschiedenen Wertstoffkreisläufen zugeteilt. Hier werden ganz verschiedene Verfahren angewandt. Dafür sind Gewicht, Materialdicke, Farbe, Transparenz, Materialdichte, Größe, usw. massgebend. Nach der Vorsortierung hilft letztlich ein ausgeklügeltes Nahinfrarotscan-System, eine möglichst gute Qualität zu erreichen.

Wie rein sind ihre recycelten Kunststoffe?

Wir erreichen eine Sortierqualität von 95 bis 98%. Bei der weiteren Bearbeitung der getrennten Kunststoff-Fraktionen wird auch noch ein Schwimm-Sink-Verfahren eingesetzt, das eine weitere Reinheitsoptimierung bringt.

Was für Schwachstellen hat ihr aktuelles System?

Es gibt immer noch diverse maschinell nicht klar definierbare Kunststoffe. Das hat damit zu tun, dass Mischkunststoff (verschiedene Fraktionen vermischt, Beispiel Käseverpackung mit PET-Schale als Basis mit Folie verschweisst) bei den Verpackungen noch sehr viel vorkommt.

Suchen Sie aktiv nach neuen Methoden, beziehungsweise Verbesserungsmöglichkeiten?

Ja natürlich, wir sind immer auf der Suche nach verbesserten Sortiersystemen. Die wichtigsten Kunststoffe (auch die von Ihnen untersuchten) sind von unserem Sortiersystem aktuell gut zu erkennen.

Falls Ja, für welchen Bereich?

Wir sind ständig (unter anderen mit der ETH) daran, die Qualität zu verbessern. Im Zentrum steht das Fernziel, die Herkunft (Shampoo-Flasche von XY) zu erkennen, diese auszusortieren und dem Hersteller XY wieder zukommen zu lassen. So kann aus der Shampoo-Flasche wieder dieselbe Shampoo-Flasche hergestellt werden. So wird der Verpackung ein zweites, drittes oder viertes Leben geschenkt und landet nicht in der Verbrennung. Unser Stofffluss wird regelmäßig nach einem EMPA-Monitoring überprüft.

Dieses Interview war für uns sehr aufschlussreich. Insbesondere hat uns das Fernziel der InnoRecycling AG der Hersteller-Erkennung zusätzlich motiviert und bestätigt, da wir der Meinung sind, dass unser Projekt das Potenzial hat, das Recycling von Polymeren auf industrieller Ebene zu verbessern.

7 Fazit

Polymere sind faszinierende und sehr nützliche Materialien, die meist aus nicht erneuerbaren Quellen stammen. Aus diesem Grund müssen Wege gefunden werden, um Kunststoffe für das Recycling zu trennen und ihre Nutzungsdauer zu verlängern. In diesem Projekt ist es uns gelungen, ein Verfahren zur Klassifizierung von Kunststoffabfällen in 4 der wichtigsten Recyclingklassen zu entwickeln und teilweise zu automatisieren. Dies erfolgte durch die Extraktion von Informationen über die Absorption und den Brechungsindex im THz-Spektrum der Polymere und die Eingabe dieser Daten in einen Algorithmus für maschinelles Lernen, der auf Support-Vector-Maschinen basiert und mit Hunderten von gelabelten Proben trainiert wurde. Weitere Arbeiten sind erforderlich, um die Einschränkungen hinsichtlich der Kunststoffarten, die wir klassifizieren konnten, zu überwinden. Da es äußerst wichtig ist, Wege zu finden, um einzigartige Kombinationen verschiedener Polymere zu trennen, hoffen wir, dass diese Methode weiter verbessert werden kann, um in grösserem Mass den Recyclingprozess zu unterstützen.

8 Danksagung

Ein ganz besonderes Dankeschön wollen wir unserem Lehrmeister und Betreuer dieser Arbeit Dominik Bachmann aussprechen. Wir möchten uns auch bei Elena Mavrona für die Unterstützung im Bereich Terahertz bedanken. Ausserdem möchten wir Peter Zolliker, Erwin Hack, Tonja Gnannt, Kalea Keith und Roman Furrer für das Korrekturlesen unserer Arbeit Danken. Anschliessend möchten wir uns bei unserem Lehrbetrieb, insbesondere unserer Abteilung Transport at Nanoscale Interfaces für die Unterstützung bedanken. Ein grosses Dankeschön an unseren Experten von Schweizer Jugend Forscht Gregory Gäumann, der uns bei der Weiterentwicklung dieser Arbeit sehr geholfen hat.

9 Ausblick

9.1 Mögliche Weiterentwicklung

Ziel unseres Projekts war es, zu zeigen, dass ein maschinelles Lernen zur Trennung von Kunststoffen anhand ihrer spektralen Informationen im THz-Bereich möglich ist. Ein kleiner Teil dieses Projekts war auch der Automatisierung einiger Teile des Prozesses gewidmet, z. B. der Dickenmessung. Dies ist uns nur in geringem Umfang gelungen, da die Messungen grösstenteils in Handarbeit durchgeführt wurden. Um das vorgeschlagene Prinzip in grösserem Umfang anwenden zu können, ist ein automatischer Ansatz erforderlich.

Die THz-Technologie beginnt gerade erst ihren Weg in die Polymerindustrie zu finden, um Qualitätsbewertungen und Prozesskontrollen durchzuführen. Wir hoffen, dass die Zukunft diese Technologie auch in andere Bereiche bringen wird. Günstigere THz-Quellen und -Detektoren machen dies bereits jetzt möglich, was ein notwendiger Schritt ist, um sie leichter zugänglich zu machen.

Wir glauben auch, dass unser Ansatz erweitert werden kann, um Verbundwerkstoffe zu kategorisieren, die viel schwieriger zu recyceln sind. Dies würde helfen, sie von den recycelbaren Kunststoffen zu trennen, um eine Verunreinigung des wiederaufbereiteten Produkts zu vermeiden. Eine weitere mögliche Anwendung unseres Ansatzes wäre die Trennung von Kunststoffen im Hinblick auf ihre verschiedenen Zusatzstoffe oder dessen jeweilige Konzentration. Dies kann eine Herausforderung sein, aber es ist möglich, dass dies die Art und Weise, wie diese leicht unterschiedlichen Kunststoffe recycelt werden, verbessern könnte.

Wie in Abschnitt [6.4](#) erwähnt, gehen viele möglicherweise relevante Informationen bei der Extraktion von Merkmalen verloren. Diese zusätzlichen Angaben könnten in einem Algorithmus nützlich sein, der alle Spektraldaten zur Klassifizierung von Materialien verwenden könnte. Ein solcher Ansatz würde eine grosse Anzahl von Messungen erfordern. Diese könnten während der Anwendung eines einfacheren Algorithmus gesammelt werden, der eine Datenbank erstellen würde, die dann zum Trainieren eines neuronalen Netzes verwendet wird. Da die gewonnenen Daten sowohl einfache als auch sehr komplexe Informationen enthalten können, könnte ein tiefes und breites Netz, wie von [\[Cheng et al., 2016\]](#) vorgeschlagen, gute Ergebnisse liefern.

Literatur

- [Bishop, 2006] Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Springer Science+Business Media, New York.
- [Boser et al., 1992] Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152. ACM Press.
- [Braun et al., 2019] Braun, J., Gahleitner, M., Grestenberger, G., Niedersüß, P., and Tranninger, C. (2019). Wachsen durch technische spezialitäten, kunststoffe.
- [Bründermann et al., 2012] Bründermann, E., Hübers, H.-W., and Kimmitt, M. F. (2012). *Terahertz Techniques*. Springer Berlin, Heidelberg.
- [Brown, 2021] Brown, S. (2021). Machine learning, explained.
- [Cheng et al., 2016] Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., Anil, R., Haque, Z., Hong, L., Jain, V., Liu, X., and Shah, H. (2016). Wide & deep learning for recommender systems.
- [Cortes et al., 2008] Cortes, C., Mohri, M., Riley, M., and Rostamizadeh, A. (2008). Sample selection bias correction theory. <https://arxiv.org/abs/0805.2775>.
- [Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*.
- [Cristianini and Shawe-Taylor, 2000] Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, Cambridge.
- [fitcecoc, 2022] fitcecoc (2022). Fit multiclass models for support vector machines or other classifiers - matlab fitcecoc - mathworks switzerland. [Online; accessed 21-September-2022].
- [Hahladakis et al., 2018] Hahladakis, J. N., Velis, C. A., Weber, R., Iacovidou, E., and Purnell, P. (2018). An overview of chemical additives present in plastics: Migration, release, fate and environmental impact during their use, disposal and recycling. *Journal of Hazardous Materials*, 344:179–199.
- [Heidenhain AG, 2022] Heidenhain AG (2022). St 3078 incremental length gauge heidenhain-specto. [Online; accessed 05-October-2022].
- [Kim, 2010] Kim, T.-h. (2010). Pattern recognition using artificial neural network: A review. In Bandyopadhyay, S. K., Adi, W., Kim, T.-h., and Xiao, Y., editors, *Information Security and Assurance*, pages 138–148, Berlin, Heidelberg. Springer Berlin Heidelberg.
- [Kindt and Schmuttenmaer, 1996] Kindt, J. T. and Schmuttenmaer, C. A. (1996). Far-infrared dielectric properties of polar liquids probed by femtosecond terahertz pulse spectroscopy. *The Journal of Physical Chemistry*, 100(24):10373–10379.
- [Krimi et al., 2016] Krimi, S., Klier, J., Jonuscheit, J., von Freymann, G., Urbansky, R., and Beigang, R. (2016). Highly accurate thickness measurement of multi-layered automotive paints using terahertz technology. *Applied Physics Letters*, 109(2):021105.
- [Kuzmenko, 2005] Kuzmenko, A. B. (2005). Kramers–kronig constrained variational analysis of optical spectra. *Review of Scientific Instruments*, 76(8):083108.
- [Lane, 2003] Lane, D. (2003). *Introduction to Statistics*. Open Textbook Library.
- [Mavrona, 2016] Mavrona, E. (2016). *Functionalised Liquid Crystals for manipulating Terahertz radiation*. PhD thesis, University of Southampton.
- [Mecademic, 2022] Mecademic (2022). Meca500 six-axis industrial robot arm. [Online; accessed 05-October-2022].
- [Mitchell, 1997] Mitchell, T. (1997). *Machine Learning*. McGraw Hill, New York.
- [Peiponen et al., 2013] Peiponen, K.-E., Zeitler, A., and Kuwata-Gonokami, M., editors (2013). *Terahertz Spectroscopy of Polymers*. Springer Series in Optical Sciences.
- [Poole et al., 1998] Poole, D., Mackworth, A., and Goebel, R. (1998). *Computational Intelligence*. Oxford University Press, New York.

- [Scheller et al., 2009] Scheller, M., Jansen, C., and Koch, M. (2009). Analyzing sub-100-um samples with transmission terahertz time domain spectroscopy. *Optics Communications*, 282(7):1304–1306.
- [Schölkopf et al., 2004] Schölkopf, B., Tsuda, K., and Vert, J.-P. (2004). A Primer on Kernel Methods. In *Kernel Methods in Computational Biology*. The MIT Press.
- [Serranti and Bonifazi, 2019] Serranti, S. and Bonifazi, G. (2019). Techniques for separation of plastic wastes. In Pacheco-Torgal, F., Khatib, J., Colangelo, F., and Tuladhar, R., editors, *Use of Recycled Plastics in Eco-efficient Concrete*, chapter 2, pages 9–37. Woodhead Publishing.
- [Sommer et al., 2018] Sommer, S., Koch, M., and Adams, A. (2018). Terahertz time-domain spectroscopy of plasticized poly(vinyl chloride). *Analytical Chemistry*, 90(4):2409–2413. PMID: 29397690.
- [Support vector machine template, 2022] Support vector machine template (2022). Support vector machine template - matlab templatesvm - mathworks switzerland. [Online; accessed 21-September-2022].
- [Wietzke et al., 2011] Wietzke, S., Jansen, C., Reuter, M., Jung, T., Kraft, D., Chatterjee, S., Fischer, B., and Koch, M. (2011). Terahertz spectroscopy on polymers: A review of morphological studies. *Journal of Molecular Structure*, pages 41–51.
- [Wikipedia, 2019] Wikipedia (2019). Libration (spektroskopie) — wikipedia, die freie enzyklopädie. [Online; Stand 11. Oktober 2022].
- [Wikipedia, 2022a] Wikipedia (2022a). Polyethylenterephthalat — wikipedia, die freie enzyklopädie. [Online; Stand 9. Oktober 2022].
- [Wikipedia, 2022b] Wikipedia (2022b). Polyvinylchlorid — wikipedia, die freie enzyklopädie. [Online; Stand 9. Oktober 2022].
- [Wikipedia contributors, 2022a] Wikipedia contributors (2022a). Parabolic reflector — Wikipedia, the free encyclopedia. [Online; accessed 9-October-2022].
- [Wikipedia contributors, 2022b] Wikipedia contributors (2022b). Polypropylene — Wikipedia, the free encyclopedia. [Online; accessed 9-October-2022].
- [Wikipedia contributors, 2022c] Wikipedia contributors (2022c). Polystyrene — Wikipedia, the free encyclopedia. [Online; accessed 9-October-2022].
- [Wikipedia contributors, 2022d] Wikipedia contributors (2022d). Steric effects — Wikipedia, the free encyclopedia. [Online; accessed 11-October-2022].
- [Zeng et al., 2021] Zeng, Z., Mavrona, E., Sacré, D., Kummer, N., Cao, J., Müller, L. A. E., Hack, E., Zolliker, P., and Nyström, G. (2021). Terahertz birefringent biomimetic aerogels based on cellulose nanofibers and conductive nano-materials. *ACS Nano*, 15(4):7451–7462. PMID: 33871983.

Anhang

Der gesamte Code und die Rohdaten sowie die verarbeiteten Daten und die Merkmaldateien sind auf GitHub zu finden:
https://github.com/cal405/thz_polymer_classification

Python Code

Imports

```
import numpy as np                                #Array manipulation
import pandas as pd                               #Dataframe writing
import os                                         #For navigating the file system
import csv                                         #Writing/Reading files
import math                                         #Perform calculations
import scipy.fftpack                                #Fast Fourier Transform
from natsort import natsorted                      #Sort files in directory
import matplotlib.pyplot as plt                     #Plot the data

#Complex refractive index extraction algorithm:
from extraction import quartz
```

Datenauswertung

```
'''

Initial setup
'''


#Where the program should search for files
#Using an example for demonstration purposes
dataPath='raw_data/20220901_foam'
colorClassified=False
processedDataDir='processed_data/20220901_foam'

#Creating figures for plots, they all open once the program reaches plt.show()
# f stands for figure, af for axis from a figure
f1, f2, f3, f4 =plt.figure(), plt.figure(), plt.figure(), plt.figure()
af1=f1.add_subplot(111)
af2=f2.add_subplot(111)
af3=f3.add_subplot(111)
af4=f4.add_subplot(111)

#Some configs for automation
dirList=natsorted(os.listdir(dataPath))
files=list()

#Config for getting sample names & thickness from dir file names
infoDict=dict()
for nnnfile in dirList:
    filepathrel=os.path.join(dataPath,nnnfile)
    files.append(filepathrel)
    splittedString=nnnfile.split('.')
    infoList=splittedString[0].split('_')

#This gives an unique color to each category depending on the filename type (first number)
if infoList[0]=='ref':
    referenceData=filepathrel
else:
    identifier=''.join(infoList[0]+'_'+infoList[2]))
    infoDict[filepathrel]={ 'name':identifier, 'thickness':int(infoList[3])}
    if int(identifier[0])==1:
```

```

        infoDict[filepathrel]['color']='red'
    elif int(identifier[0])==2:
        infoDict[filepathrel]['color']='magenta'
    elif int(identifier[0])==4:
        infoDict[filepathrel]['color']='green'
    elif int(identifier[0])==5:
        infoDict[filepathrel]['color']='blue'
    else:
        infoDict[filepathrel]['color']='black'

''''
First figure (Electrical Signal vs. Time [ps])
'''

#Get pure signal
dataSpectrum=dict()
counter=0
for nfile in infoDict:
    tArray = [list(),list()]
    with open(nfile) as csvfile:
        csvreader = csv.reader(csvfile, delimiter = ',')
        next(csvreader)
        for row in csvreader:
            tArray[0].append(float(row[0]))
            tArray[1].append(float(row[1]))
    dataSpectrum.update({nfile:tArray})
    if colorClassified:
        af1.plot(dataSpectrum[nfile][0],dataSpectrum[nfile][1],
                  linewidth=2.0, label=nfile, linestyle='--', color=infoDict[nfile]['color'])
    else:
        af1.plot(dataSpectrum[nfile][0],dataSpectrum[nfile][1],
                  linewidth=2.0, label=nfile, linestyle='--')
#Settings for plot
af1.set_xlim([2370,2450])
af1.set_ylabel('E.F.', fontsize=30)
af1.set_xlabel('Time [ps]', fontsize=30)
plt.draw()

''''
Fast Fourier Transform up to 3 THz
''''

#Config for the FFT
tau=np.array(dataSpectrum[files[0]][0])
dt=tau[1400]-tau[0]
df=1/dt
f=np.multiply(df,np.array(range(0,1401)))

#Plot the FFT
for nnfile in infoDict:
    if colorClassified:
        af2.plot(f,abs(scipy.fftpack.fft(dataSpectrum[nnfile][1])),
                  linewidth=2.0, label=nnfile, linestyle='--', color=infoDict[nnfile]['color'])
    else:
        af2.plot(f,abs(scipy.fftpack.fft(dataSpectrum[nnfile][1])),
                  linewidth=2.0, label=nnfile, linestyle='--')
#Settings for plot
af2.set_xlim([0,3])
af2.set_ylabel('Spectrum', fontsize=30)

```

```

af2.set_xlabel('Frequency (THz)', fontsize=30)
plt.draw()

'''

Algorithm comes into play to calculate the refractive index

'''


#Define here your sample files with thickness in mm
for key in infoDict:
    realPart, imaginaryPart, frequency =quartz(
        referenceData,key, (float(infoDict[key]['thickness'])/1000.0)
    )
    infoDict[key]['imaginaryPart']=imaginaryPart
    infoDict[key]['frequency']=frequency
    infoDict[key]['realPart']=realPart

#Calculating the absorption
lamda=3.0e8/frequency
absorption=-4*math.pi*imaginaryPart/(lamda*100)
infoDict[key]['absorption']=absorption

#Saving results to file
bareFileName=key.split('\\')
newFileName=os.path.join(processedDataDir,bareFileName[-1])
'''with open(newFileName,'w',newline='') as proccsvfile:
    csvwriter=csv.writer(proccsvfile,delimiter=',')
    csvwriter.writerow(['refractiveIndex','absorption'])
    for row in range(len(frequency)-1):
        csvwriter.writerow([realPart[row+1],absorption[row+1]])
with open(newFileName,'w',newline='') as proccsvfile:
    csvwriter=csv.writer(proccsvfile,delimiter=',')
    csvwriter.writerow(['frequency','refractiveIndex','absorption'])
    for row in range(len(frequency)-1):
        csvwriter.writerow([frequency[row+1]*1e-12,
                           realPart[row+1],absorption[row+1]])'''

#Drawing refractive index plot
for key in infoDict:
    if colorClassified:
        af3.plot((infoDict[key]['frequency'])*1e-12,(infoDict[key]['realPart']),
                  linewidth=2.0, label=infoDict[key]['name'], linestyle='--',
                  color=infoDict[key]['color'])
    else:
        af3.plot((infoDict[key]['frequency'])*1e-12,(infoDict[key]['realPart']),
                  linewidth=2.0, label=infoDict[key]['name'], linestyle='--')

#Settings for plot
af3.set_xlim([0,3])
af3.set_ylim([-1,3])
af3.set_ylabel('Refractive index', fontsize=30)
af3.set_xlabel('Frequency (THz)', fontsize=30)
af3.legend(loc='upper right', prop={'size':10})
plt.draw()

#Drawing absorption plot
for key in infoDict:
    if colorClassified:
        af4.plot((infoDict[key]['frequency'])*1e-12, infoDict[key]['absorption'],
                  linewidth=2.0, label=infoDict[key]['name'], linestyle='--')

```

```

        linewidth=2.0, label=(infoDict[key]['name']), linestyle='--',
        color=infoDict[key]['color'])
    else:
        af4.plot((infoDict[key]['frequency'])*1e-12, infoDict[key]['absorption'],
        linewidth=2.0, label=(infoDict[key]['name']), linestyle='--')
#Configure your absorption plot
af4.set_xlim([0,3.0])
af4.set_ylim([-10,200])
af4.set_ylabel('Absorption (1/cm)', fontsize=30)
af4.set_xlabel('Frequency (THz)', fontsize=30)
af4.legend(loc='upper right', prop={'size':10})
plt.draw()

plt.show()

```

Merkalsextraktion

```

def feature_extraction(pathGetData='ml/train/data', pathOutput='ml/train/featureFile.csv'):
    """
    This function reads the whole dir in pathGetData and dumps the information
    that it gets from the filename (plastic type) and the features specified in
    the body of the function. Saves file at pathOutput.
    """

#Initial config
filenames=natsorted(os.listdir(pathGetData))
labels=list()

#Features to extract
nAt2_3THz=list()
alphaAt2_3THz=list()
alphaStdDevfrom0_5to2_5THz=list()
nStdDevfrom0_5to2_5THz=list()
alphaAvgfrom0_5to1_5THz=list()
alphaAt1_3THz=list()

#Extracting features
for filename in filenames:
    #Loading data into memory
    sampleName=filename.split('.')[2]
    if sampleName[0]=='1':
        labels.append('PET')
    elif sampleName[0]=='5':
        labels.append('PP')
    elif sampleName[0]=='2':
        labels.append('HDPE')
    elif sampleName[0]=='4':
        labels.append('LDPE')
    elif sampleName[0]=='6':
        labels.append('PS')
    elif sampleName[0]=='3':
        labels.append('PVC')
    datafram=np.genfromtxt(pathGetData+'/'+filename, delimiter=',')
    datafram=dataframe[~np.isnan(dataframe).any(axis=1), :]

    #Here we define where we want to get the data from
    # n and alpha at 2.3THz
    findSpot2_3THz=np.where(dataframe[:,0]==2.2304685965489504)
    nAt2_3THz.append(dataframe[findSpot2_3THz[0],1][0])

```

```

alphaAt2_3THz.append(dataframe[findSpot2_3THz[0],2][0])
# n and alpha from 0.5 to 2.5 THz
startSpot0_5THz=np.where(dataframe[:,0]==0.500817154756336)
endSpot2_5THz=np.where(dataframe[:,0]==2.50042125801517)
alphaStdDevfrom0_5to2_5THz.append(
np.std(dataframe[startSpot0_5THz[0]:endSpot2_5THz[0],2]))
)
nStdDevfrom0_5to2_5THz.append(np.std(
dataframe[startSpot0_5THz[0]:endSpot2_5THz[0],1]))
)
# alpha from 1.5 to 2.5 THz
endSpot1_5THz=np.where(dataframe[:,0]==1.5000084537580014)
alphaAvgfrom0_5to1_5THz.append(np.mean(
dataframe[startSpot0_5THz[0]:endSpot1_5THz[0],2]))
)
# alpha at 1.3 THz
findSpot1_3THz=np.where(dataframe[:,0]==1.3002923444832186)
alphaAt1_3THz.append(dataframe[findSpot1_3THz[0],2][0])

#Loading data into temp dictionary
data={'Category':labels,
'n at 2_3THz':nAt2_3THz,
'alpha at 2_3THz':alphaAt2_3THz,
'alphaStdDevfrom0_5to2_5THz':alphaStdDevfrom0_5to2_5THz,
'nStdDevfrom0_5to2_5THz':nStdDevfrom0_5to2_5THz,
'alphaAvgfrom0_5to1_5THz':alphaAvgfrom0_5to1_5THz,
'alpha at 1_3THz':alphaAt1_3THz}

#Dumping all data to file
df = pd.DataFrame.from_dict(data)
df.to_csv(pathOutput, index=False, header=True, sep=',')

```

MATLAB Code

Machine Learning for THz classification of Polymers

Import Data (feature file)

```

opts=detectImportOptions("featureFile.csv");
train=readtable("featureFile.csv",opts, ...
    "ReadVariableNames",false);
test=readtable("featureFileTest.csv",opts, ...
    "ReadVariableNames",false);

```

Visualize Training Data

```

colors=[0.4940 0.1840 0.5560;1 1 0;0.3010 0.7450 0.9330;0.4660 0.6740 0.1880];
gscatter(train.nAt2_3THz,train.alphaAt2_3THz,train.Category,colors)
xlabel('n @ 2.3THz')
ylabel('alpha @ 2.3THz')

```

```

gscatter(train.nAt2_3THz , ...
    train.nStdDevfrom0_5to2_5THz, train.Category,colors)
xlabel('n @ 2.3THz')
ylabel('n StdDev from 0.5 to 2.5THz')

```

Create and train SVM model

```
[model,validationDataAcc]=trainClassifier(train);
validationDataAcc
```

3D Visualization

```
figure
set(gcf,'Visible','on')
scatter3(train.nAt2_3THz,train.alphaAt2_3THz,train.nStdDevfrom0_5to2_5THz, ...
10,categorical(train.Category),'filled')
xlabel('n @ 2.3THz'); ylabel('alpha @ 2.3THz'); zlabel('n StdDev from 0.5 to 2.5THz');
```

Confusion Matrix of the Training set

```
predictionsTrainSet=modelfinal.predictFcn(train);
figure
confusionchart(categorical(train.Category),predictionsTrainSet, ...
'Title','Confusion Matrix - Training Set')
```

Test model

```
tic
predictionsTestSet=modelfinal.predictFcn(test);
toc

TestAcc=sum(predictionsTestSet==test.Category ...
)/length(predictionsTestSet)

figure
confusionchart(categorical(test.Category),predictionsTestSet, ...
'Title','Confusion Matrix - Test Set')
```

Functions

```
function [trainedClassifier, validationAccuracy] = trainClassifier(trainingData)

inputTable = trainingData;
predictorNames = {'nAt2_3THz', 'alphaAt2_3THz', 'alphaStdDevfrom0_5to2_5THz', ...
'nStdDevfrom0_5to2_5THz', 'alphaAvgfrom0_5to1_5THz', 'alphaAt1_3THz'};
predictors = inputTable(:, predictorNames);
response = inputTable.Category;
isCategoricalPredictor = [false, false, false, false, false, false];

template = templateSVM(... 
    'KernelFunction', 'gaussian', ...
    'PolynomialOrder', [], ...
    'KernelScale', 2.4, ...
    'BoxConstraint', 1, ...
    'Standardize', true, ...
    'SaveSupportVectors', true);
classificationSVM = fitcecoc(... 
    predictors, ...
    response, ...
    'Learners', template, ...
    'Coding', 'onevsone', ...
    'ClassNames', categorical({'PET'; 'PP'; 'PS'; 'PVC'}));

% Create the result struct with predict function
```

```

predictorExtractionFcn = @(t) t(:, predictorNames);
svmPredictFcn = @(x) predict(classificationSVM, x);
trainedClassifier.predictFcn = @(x) svmPredictFcn(predictorExtractionFcn(x));

% Add additional fields to the result struct
trainedClassifier.RequiredVariables = {'alphaAt1_3THz', 'alphaAt2_3THz',...
    'alphaAvgfrom0_5to1_5THz', 'alphaStdDevfrom0_5to2_5THz', 'nAt2_3THz',...
    'nStdDevfrom0_5to2_5THz'};
trainedClassifier.ClassificationSVM = classificationSVM;

% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
inputTable = trainingData;
predictorNames = {'nAt2_3THz', 'alphaAt2_3THz',...
    'alphaStdDevfrom0_5to2_5THz', 'nStdDevfrom0_5to2_5THz',...
    'alphaAvgfrom0_5to1_5THz', 'alphaAt1_3THz'};
predictors = inputTable(:, predictorNames);
response = inputTable.Category;
isCategoricalPredictor = [false, false, false, false, false, false];

% Perform cross-validation
partitionedModel = crossval(trainedClassifier.ClassificationSVM, 'KFold', 5);

% Compute validation predictions
[validationPredictions, validationScores] = kfoldPredict(partitionedModel);

% Compute validation accuracy
validationAccuracy = 1 - kfoldLoss(partitionedModel, 'LossFun', 'ClassifError');
end

```