

Wafer-to-wafer variations

Ein datengetriebener Ansatz zum Erkennen von Wafer-Defekten und -Anomalien bei der Stichprobennahme in der industriellen Fertigung

Norman Bauersfeld*
u37724@hs-harz.de
Goslar, Deutschland

Zusammenfassung

Silizium-Wafer dienen als eine Grundlage zur Herstellung von Mikrochips und elektronischen Bauteilen. Durch den vielschichtigen Fertigungsprozess entstehen verschiedene als Fehler klassierte Oberflächenphänomene, die mithilfe von datengetriebenen Verfahren frühst möglich erkannt werden müssen. Mit dieser Arbeit wird eine auf Algorithmen des maschinellen Lernens basierende mehrstufige Analyse anhand von ex-situ Raster-Elektronenmikroskop-Aufnahmen entwickelt und deren Funktionsfähigkeit anhand eines im industriellen Umfeld gewonnenen Datensatzes sogenannter binärer Waferkarten demonstriert. Es wird veranschaulicht, welche Möglichkeiten gegeben sind, Anomalien zu den bereits bekannten Fehlerbildern des Datensatzes zu ermitteln, und in neue Fehlerklassen einzuordnen. Zusammenfassend werden die Ergebnisse der Analyse einer Diskussion unterzogen und Möglichkeiten der Erweiterung der Analyse aufgezeigt.

Keywords: Wafer-Defekt, SEM, CNN, t-SNE, DBSCAN, Ausreißer

1 Einleitung

Im Jahre 1959 bestanden Chips aus einem einzigen Transistor, heute enthalten sie mehr als eine Million Bauelemente [31]. Chips sind mikroelektronische Bauelemente auf Halbleiter- bzw. Siliziumbasis. Die Aufgabe, Wafer¹ aus Silizium für die Herstellung von mikroelektronischen Bauelementen, fehlerfrei und ertragreich zu produzieren, wird als herausfordernd angesehen. Die Vielzahl der Bearbeitungsschritte, um von einem Ingot² zum Wafer zu gelangen, sowie die eingesetzten Produktionsmittel bedingen das quellen-spezifische, auf einen Messpunkt bezogene, Ermitteln von Wafer-Produkt- und Produktionseigenschaften. Herstellerwissen erlaubt die charakteristische Einstufung einiger Eigenschaften als Materialfehler, welche sich optisch, visuell/bildlich als Einschlüsse, Klumpen, Ringe oder Kratzer oder deren Kombinationen darstellen lassen und Verunreinigungen an oder auf Betriebsmitteln³, Fehleinstellungen von Produktionsparametern oder Betriebsmittelverschleiß melden. Zahlreiche Veröffentlichungen über visuelle Abweichungen auf Waferkarten, den Wafer-Defekten, deren Erkennen, Kennzeichnen und Bewerten sowie daraus Rückschlüsse auf Material- und Prozesszustände zu ziehen, weisen das rege Interesse nach, dieses Thema im Run-to-Run Steuerungsprozess lösbar und industriell praktikabel zu gestalten, vgl. [32]. Im Artikel [31] wird bereits im Jahre 1987 darauf hingewiesen, dass bei der Chipherstellung mit einer zunehmenden Miniaturisierung bei gleichzeitiger Erhöhung der Schaltkreisintegrationsdichte zu rechnen ist und dieser Trend

dauerhaft anhalten wird. Ein guter Grund, sich mit dem Thema der prozessoptimierten und ertragssteigernden Herstellung von Silizium-Wafern zu beschäftigen. Der technische Einstieg in das Thema der Raster-Elektronenmikroskopie (SEM⁴) zum Erzeugen der Waferkarten wird durch [27] möglich. Die Autoren in [41] zeigen exemplarisch einen Versuchsaufbau für die Fehlererkennung von unpolierten Silizium-Wafern in der digitalen Shearografie⁵. So wird anschaulich die Entstehung von Illustrationen binärer Waferkarten zur Klassierung beschrieben. Die Autoren in [12] untersuchen statistisch Fehlerdichten auf Waferkarten und schlagen neuartige statistische Analysen in der Defekt-Metrologie vor. Der Artikel wird nutzbringend für die unten beschriebene Datenvorbereitung verwendet. Der Autor in [32] beschreibt die Run-to-Run (RtR) Steuerung als eine Form der adaptiven modellbasierten Prozesssteuerung, welche auf entsprechende Prozessumgebungen angepasst, in der Halbleiterfertigung Einsatz findet. Die Auswertung von Waferkarten liefert an unterschiedlichen Messpunkten unterschiedliche Aussagen über das Prozessgeschehen und hat damit auch unterschiedliche Implikationen zur Folge. Für die hier angedachte datengetriebene Auswertung mittels eines neuronalen Netzes sind die Ausführungen in [22] oder [6] von Interesse. Sie gehen auf Probleme beim Trainieren von CNN⁶ mit nicht ausbalancierten Daten ein. Um diesen Umstand zu beheben, werden in [20], [30], [9] oder [25] Methoden und Verwendung von Autoencodern oder GAN⁷ vorgestellt. In [49] stellen die Autoren für die Anreicherung spezielle bildmanipulierende Methoden, wie Vertauschen (engl. reverse), Verschieben und Rotieren (engl. translate) und Kombinieren (engl. combine) auf vorhandenen Daten vor. Die Auswahl für den entwickelten Prototypen fiel auf einen ausbalancierten Datensatz. Damit werden die dort beschriebenen Probleme umgangen. In [9] wird ein automatisches Klassifizierungssystem entwickelt, welches Wafer-Defekte anhand von SEM-Bildern klassiert, hauptsächlich aber mit der Zielstellung, dass Hersteller entscheiden können, ob bspw. ein Wafer reparabel ist. Für die Fehlerklassifizierung selbst wird ein CNN vorgeschlagen, welches ohne zusätzliche Merkmalsextraktionsalgorithmen Aussagen zu unbekannten Klassen treffen kann. Diese Idee wird aufgegriffen und in Form von Betrachtungen zur Erkennung unbekannter Fehlertypen zur Geltung kommen. Darüber hinaus werden in [43] kombinierte Defektmuster auf Waferkarten erkannt und mithilfe eines DCNN⁸ aufgeschlossen. Die Autoren stellen den Forschungsdatensatz zur Verfügung [44], der in dieser Arbeit Verwendung findet. Eine weiterführende Einarbeitung in Algorithmen und Kennzahlen des maschinellen Lernens

*Studiengang bbgf. Data Science (M. Sc.), Matrikel 30315, Fachbereich A/I, Hochschule Harz, Friedrichstr. 57-59, 38855, Wernigerode

¹Wafer, engl. für dünne Scheibe, bezeichnet i.d.R. kreisrunde Scheiben als Substrat/Grundplatte

²Ingot, engl. für Barren, bezeichnet einen Block aus Silizium Halbleitermaterial

³Betriebsmittel, bezeichnen Materialien zur Bearbeitung sowie Maschinen

⁴SEM, Abk., engl. Scanning Electron Microscopy

⁵Shearography, engl. für Laser Speckle Shearing Interferometrie, kohärent optisches Messverfahren

⁶CNN, Abk., engl. Convolutional Neural Net

⁷GAN, Abk., engl. Generative Adversarial Network

⁸DCNN, Abk., engl. Deformable CNN

gelingt u. a. für das verwendete t-SNE mit [42], dem DBSCAN⁹ mit [16], für die ROC¹⁰ [17] oder allgemein zum Thema maschinelles Lernen und die Auslegung von CNN mit [3]. Diese Kenntnisse werden für die Modellbildung und in der Ausreißer-Erkennung bzw. dem Erkennen von unbekannten Fehlerklassen von Nutzen sein. Eine Fallstudie aus [46] zeigt, dass ein struktureller Aufbau nach CRISP-DM¹¹ auch für die datengetriebene Defekterkennung von Silizium Trägerscheiben geeignet ist.

In verschiedenen aufeinander folgenden, partiell wiederholbaren, Arbeitsschritten wird eine Aufgabe zur Datenanalyse zu einem anwendbaren prototypischen Software-Demonstrator überführt. Das Verständnis für die Aufgabenstellung ermöglicht die Datenakquise und Vorverarbeitung, gefolgt von der Ausarbeitung eines Modells zur Datenauswertung. Die erzielten Ergebnisse werden plausibilisiert und bewertet. Die bereits erwähnte SEM, mit Hilfe derer der im Experiment analysierte Datensatz erstellt wurde, verwendet im Gegensatz zu optischen Messverfahren anstelle eines Lichtstrahls einen fokussierten Elektronenstrahl, um von einer Probe, den Oberflächen der Wafer-Scheiben, ein informatives Abbild über Zusammensetzung und Struktur in sehr hoher Auflösung zu erhalten, [27]. Stichprobenartige ex-situ Kontrollen der RfR-Verfahrenssteuerung inspizieren unbeschichtete Wafer elektronenmikroskopisch. Die Messgrößen dienen als Fehlerkoordinaten zum Erstellen einer binären Waferkarte für eine weitere Datenanalyse. Die SEM erlaubt die monochromatische zerstörungsfreie Aufnahme von Materialstrukturen durch gezielten Beschuss einer Probe mit einem gebündelten Elektronenstrahl. Die mit der Materie der Probe entstehenden Wechselwirkungen erzeugen elektrisch detektierbare Signale von Sekundärelektronen, Röntgenstrahlen, Auger-Effekt-Elektronen, Elektronen aus der Rückstrahlung und elektromagnetischer Oberflächenrückstrahlung zur Struktur eines Wafers und werden in die spezifischen Informationen in Bild-/Bitmustern, die sogenannten Waferkarten, umgewandelt.

2 Modell

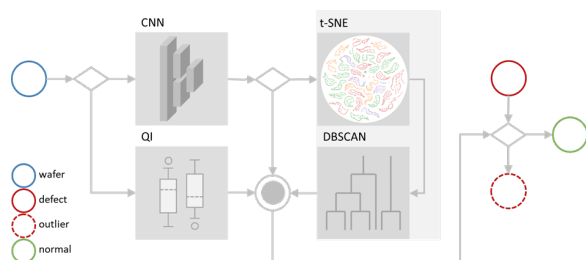


Abbildung 1. Klassifikationspipeline

In vielen industriellen Bereichen finden Modelle des maschinellen Lernens bereits Anwendung, und kommen dort insbesondere bei der Bilderkennung oder der Extraktion von kompakten Indikatoren aus mehrdimensionalen Messdatenreihen zum Einsatz, vgl. [8] oder [18]. Motivierend dabei ist die Möglichkeit der softwaretechnischen prototypischen Auslegung eines solchen Modells. Die hier nach der Abb. 1 modellierte, später in [2] implementierte, und datengetriebene Klassifikation arbeitet nach inhaltlich autonom

sequenziell ausgelegten Schritten. Ein Teil-Modell-Schritt der Klassifikationspipeline meldet, je nach späterer Auslegung, Normal-, Defekt- und Ausreißer-Typen. Jedem Teil-Modell wird der vollständig gelabelte Trainingsdatensatz, evtl. Ausgaben aus vorangegangenen Schritten, vorgelegt. Damit können Ergebnisse miteinander verglichen und validiert werden.

Schritt 1, QI: Eine erste Häufigkeitsanalyse ermittelt den prozentualen mittleren Anteil von Defektpunkten zu Normalpunkten auf binären Waferkarten. Vergleiche über den Quartilsabstand, die Lage des Mittelwertes oder die Verteilung der Normal- und Defektpunkten über robuste Streuungs- und Lagemaße, vgl. [5] und [35], wie die Spannweite, geben Hinweise auf mögliche starke Abwandlungen der Typen untereinander oder Ausreißer innerhalb eines Typs.

Schritt 2, CNN: Der als CNN ausgelegte neuronale Klassifikator findet Einsatz für das eigentliche Erkennen der kritischen Defekttypen und wird so durch entsprechendes Training angelernt, dass unbekannte Test-Defekttypen ordnungsgemäß und mit hinreichend hoher Güte vorherzusagen sind. CNN können die Topologie von Bildern effektiv erfassen, kontextbezogen speichern und eine große Anzahl von Klassen unterscheiden, vgl. [28] oder [39]. Das CNN-Modell bestimmt maßgeblich die Güte der Vorhersage bei der Präsentation unbekannter, zu klassierender Daten. Das hier experimentell durch Ergebnisvergleich begünstigte Modell verfügt über eine, auf dem Datensatz [44] erlernte Struktur. Die Eingabebildinformation wird zweimalig gefaltet und komprimiert (engl. convolution and pooling). Die zweidimensionalen Bildkoordinaten werden darauf hin in einen eindimensionalen Merkmalsvektor überführt (engl. flatten). Ein Anteil des Erlernten wird wieder verworfen (engl. dropout, [19], [40]), um Überanpassung zu vermeiden und die Struktur robuster und genereller zu gestalten. Die Ausgabe realisiert ein 38-stelliger Layer (engl. dense; fully connected); jede Stelle, entspricht einem Defekttyp. Jeder Defekttyp wird durch die Ermittlung der Stelle des maximalen Wertes vorhergesagt. Die technische Auslegung der Layer ist nach Tabelle 1. Dabei beziehen sich die Modelldimensionen bereits auf den später im Experiment verwendeten Datensatz aus Abschnitt 4.

Layer	Beschreibung
Input	Eingabe der Bildinformation 52x52
Conv2D ¹² (ReLU ¹³)	Faltung auf 50x50 mit 32 Filtern
MaxPooling2D ¹⁴	Maximalwertkomprimat auf 25x25 Karten
Conv2D(ReLU)	Faltung auf 23x23 mit 64 Filtern
MaxPooling2D	Maximalwertkomprimat auf 11x11 Karten
Flatten	11x11 Karten, 64 Filter in 1d-Vektor auflösen
Dense(ReLU)	Up-Sampling auf 128 Stellen
Dropout	Zwischenergebnisse verwerfen, s.o.
Dense(SoftMax ¹⁵)	38-stellige Ausgabe

Tabelle 1. Technische Auslegung des CNN

Die Differenz zwischen Labels und Vorhersagen wird mittels des Kreuzentropieverlustes für kategoriale Daten (engl. categorical cross entropy loss) berechnet. Das Netz wird während der Lernphase, und während der Präsentation der Trainingsdaten, versuchen, diesen Verlust zu minimieren. Sind zwei oder mehr Ausgabebezeichnungen möglich, entsteht bei jeder verketteten Ergebnismittlung (engl. propagation) des Netzes ein Ausgabelabel als Codierungswert für ein One-Hot-Labeling (OHL) im Intervall [0, 1].

⁹DBSCAN, Abk., engl. Density Based Spacial Clustering of Applications with Noise

¹⁰ROC, Abk., engl. Receiver Operating Characteristic

¹¹CRISP-DM, Abk., engl. CROSS Industry Standard Process for Data-Mining

Durch eine Suche nach der Stelle mit dem maximalen Wert wird eine Aussage zur Kategorie, zum propagierten Defekttyp, getroffen.

Das überwachte Trainieren des neuronalen Netzes und die adaptive Anpassung der Lernrate erfolgt durch einen Optimierer im stochastischen Gradienten-Abstiegsverfahren (engl. stochastic gradient descent, SGD). Dieser passt die Gewichte des Netzes derart an, dass Falschaussagen zwischen der Eingabe von Proben als Defekttypen und prognostizierten Ausgaben minimiert werden. Dabei erfolgt jedoch die Berechnung des Gradienten nicht global über den ganzen Datensatz, sondern nur über einen zufällig gewählten Teil pro Lernphase. Es wird vorausgesetzt und kann gezeigt werden, dass die zufällig gewählten Daten der Verteilung der Grundgesamtheit entsprechen und der angenäherte Gradient zu einer Netzkonvergenz führt. Das wird unterstützt durch eine Adaption der Lernrate während des Lernens; initial mit 0.01 und einer stufenweisen Absenkungsrate von 0.98 nach der zweifachen Präsentation des Datensatzes.

Die Hyperparameter des CNN werden experimentell anhand des Konvergenzverhaltens beim Training ermittelt. Nimmt die Lernrate zu schnell ab, dauert es mitunter sehr lange, bis ein, evtl. nur lokales, Optimum erreicht wird. Nimmt die Lernrate zu langsam ab, wird ein Optimum evtl. niemals erreicht, vgl. [19]. Das SGD-Verfahren wurde eingeführt, um bei zunehmender Größe von Datensätzen, die Anzahl der Konvergenzberechnungen und damit die Rechenzeit und Trainingszeit zu reduzieren. [4] geht hier weiterführend auf etwaige Fehlerquellen ein und beschreibt anschaulich, wie sich der Vorhersagefehler zusammensetzt und weiter reduziert werden kann. Weitere Parameter sind die Anzahl der zu trainierenden Epochen mit 25 und die Batch¹⁶-Größe mit 64.

Schritt 3, t-SNE: Der t-SNE visualisiert die durch das CNN gelieferten 38-dimensionalen Vorhersagen in einen 2-dimensionalen Merkmalsraum durch eine nicht lineare Dimensionsreduktion. Ziel ist es, eine ähnliche bedingte Wahrscheinlichkeitsverteilung von Datenpunkten, wie im hochdimensionalen Raum zu erreichen. Dabei sollen lokale strukturelle Zusammenhänge erhalten bleiben. Zur Optimierung der Verteilung minimiert t-SNE die Kullback-Leibler-Divergenz (KLD, engl. information gain), vgl. [42] und [29]. Die KLD ist ein Maß für die Unterschiedlichkeit zweier Verteilungen, insbesondere hier als Differenzmaß für die Reproduktion der angenommenen Gaußschen Verteilung des originalen Datenraumes auf die t-Student-Verteilung des gefalteten Datenraumes. Als experimentell ermittelte Parameter werden für die Berechnungen nach [34] die Durchmischung (engl. perplexity) mit 20, die maximale Anzahl von Versuchen mit 1000 sowie die Clusterdichte im niedrigdimensionalen Raum (engl. embedded space) von 10 herangezogen.

Schritt 4, DBSCAN: Der in Schritt 3 durch t-SNE ermittelte Merkmalsraum (engl. embeddings) wird mittels defekttypenbasierter DBSCAN's [16] so standardskaliert ausgewertet, dass dichtverbundene Stichproben innerhalb jedes Typen sowohl in Trainings- als auch in Testdaten gefunden werden. Die Proben gelten als dicht, wenn ihr (euklidischer) Abstand ein gegebenes Maß unterschreitet. Ist eine Probe mit einer gegebenen Anzahl von dichten Proben verkettet, gilt sie als Kernprobe. Proben, welche durch dieselben Kernproben miteinander verbunden sind, bilden einen Defekttyp (allg., engl. cluster). Proben außerhalb von Verbänden gelten als Rauschobjekte bzw. Ausreißer. Die Anzahl der mindestens vorhandenen nächsten Nachbarn, damit eine Stichprobe eine Kernprobe bilden

kann, ist mit 15 angegeben. Der Abstand, unter dem Proben als dicht verbunden gelten, liegt im standardskalierten Merkmalsraum bei 0.5. DBSCAN muss nicht bekannt sein, wie ein vorgegebener Datensatz inhaltlich strukturiert ist. Der Algorithmus arbeitet exakt in Bezug auf die Definition von Dichteverbund und Rauschen und ermittelt bei der Ausbildung neuer nicht zuzuordnender Gruppen evtl. neue Defekttypen.

3 Evaluation

Die in der Abb. 1 gezeigten Schritte QI¹⁷, CNN und DBSCAN erkennen Defekttypen und Ausreißer.

Die QI (*Schritt 1*) unterteilt anhand von Auswertungen der Häufigkeitsanteile aus Defekt- und Normalpunkten. Zudem ergeben Streuungs- und Lagemaße Hinweise auf charakteristische Merkmale. Es wird davon ausgegangen, dass bspw. die außerhalb der Antennen (engl. whisker) liegenden Stichproben als Ausreißer (engl. fliers) markiert werden können. Kastengrafiken über alle Defekttypen visualisieren das anschaulich. Das CNN kann 38-stellige Ausgangslabel liefern, welche sich in keine bekannten Kodierungen einordnen lassen und definiert diese als neuen Defekttyp oder Ausreißer. DBSCAN detektiert Stichproben auf t-SNE Embeddings, welche weit außerhalb eines dichtebasierten Verbandes liegen. So ist sichergestellt, dass die Klassifikationspipeline neben Normaltyp und den benannten Defekttypen, Ausreißer an unterschiedlichen Stellen der Teilmodelle ordentlich separiert.

Die CNN-Modellbewertung (*Schritt 2*) erfolgt durch Metriken einer Klassifikation, den Gütemaßen nach [19], welche jeweils in der Lernphase pro Epoche für Training und Validierung, und in der Anwendungsphase bei der Vorhersage, berechnet werden. Gleichbedeutend wird die durch den Optimierer minimierte Kreuzentropieverlustfunktion auf kategorialen Daten mit OHL mit realwertigen Stellen aufgezeichnet. Die Bestimmung des prozentualen Verhältnisses der Vorhersagen zu den wahren Werten erfolgt im Sinne einer kategorialen Genauigkeit (engl. categorical accuracy) und bezieht sich auf die OHL der Kategorien der Vor- und Ausgaben und wird verwendet, da die Defekttypen (Kategorien) in gleichen Anteilen trainiert werden. Um einen Eindruck vom Anteil korrekter positiver Vorhersagen zu bekommen, wird die Relevanz (engl. precision) berechnet. Es wird eine allgemeine Aussage zur Trefferquote (engl. recall), und somit zum Anteil positiver Vorhersagen bzw. Sensitivität (engl. sensitivity) erreicht. Das harmonische Mittel aus Relevanz und Sensitivität bestimmt das $F_{\beta=1}$ Maß. Es findet Beachtung, weil das Maß nur dann hoch ist, wenn auch die einfließenden Maße einen hohen Wert erreichen. Mit der ROC¹⁸-Kurve wird die Richtig-Positiv-Rate (engl. true positive rate), gegen die Falsch-Positiv-Rate (engl. false positive rate), abgetragen. Das Maß ist dann die Fläche unter der ROC-Kurve (engl. area under curve). Ein Klassifikator wird als gut bezeichnet, wenn das Maß der Fläche gegen 1 strebt. Für eine Zuverlässigkeitsaussage zur Genauigkeit der Vorhersagen wird Cohen's Kappa [11] aus den beobachteten und den erwarteten Übereinstimmungen berechnet; völlige mit 1, zufällige gegen 0.

Eine weitere Möglichkeit, die Güte eines neuronalen Lernalters zu bestimmen, ist das Aufstellen einer Konfusionsmatrix. Jede Zeile steht für eine tatsächliche Klasse, jede Spalte für die vorhergesagte. An den Kreuzungspunkten wird der jeweilige Anteil aus dem

¹⁶Stichprobenanzahl für die Aktualisierung der Modellparameter

¹⁷QI, Abk., engl. Quality Inspection

¹⁸ROC, Abk., engl. Receiver Operating Characteristic

Datensatz eingetragen. Die Werte der Hauptdiagonalen sollten für einen guten Klassifikator die am höchsten bewerteten Kennziffern tragen. In der experimentellen Auslegung wird die Matrix über die Trainings- und Testdaten gebildet und anhand der diagonalen Werte begutachtet. Um ein schlüssiges Bild über die Fehlvorhersagen und deren Ursache zu erhalten, werden genau die Illustrationen aus dem Datensatz extrahiert und für einen augenscheinlichen Vergleich vorgeschlagen, die als Ausreißer postuliert werden, also Stichproben deren Defektbild ungewöhnlich ist. Die augenscheinliche Einordnung erfolgt dann entweder in tatsächliche Ausreißer innerhalb oder in einen neuen Defekttyp. Ein neuer Defekttyp wird eine Neubewertung der aktuellen Prozesssituation erforderlich machen.

Die Clustergütebestimmung der DBSCAN (*Schritt 4*) erfolgt durch das Silhouette-Maß, [36], (engl. auch silhouette score) und errechnet sich aus dem Mittel aller Silhouette Koeffizienten der Stichproben. Der Silhouette Koeffizient ergibt sich aus der mittleren Intra-Cluster-Distanz eines Punktes zu seinen Mit-Cluster-Punkten und der mittleren Nächsten-Cluster-Distanz eines Punktes eines Clusters zu den Cluster-Punkten des am nächsten liegenden Clusters. Die Werte 1, 0, -1 bedeuten, ein Punkt liegt weit entfernt zu einem nächsten Cluster und ist gut klassifiziert, ein Punkt liegt in der Nähe der Grenzlinie zwischen seinem und dem nächsten Cluster und, ein Punkt liegt in einem falschen Cluster. Das Davies-Bouldin-Maß, [13], definiert die mittlere Ähnlichkeit zwischen Clustern durch Vergleich der Clusterabstände untereinander und der Clusterausdehnung. Je kleiner der Wert, desto besser ist das Clusterergebnis einzuordnen. Das Calinski-Harabasz-Maß, [7], (engl. variance ratio criterion) betrachtet Heterogenität und Homogenität. Heterogenität verdeutlicht die Abstände zwischen den Clusterzentren; je größer die Distanz, desto besser die Trennung. Homogenität ermittelt sich aus der Quadratsumme der Abstände von Cluster-Punkten zu ihrem Zentrum; je kleiner die Summe, desto homogener sind die Cluster. Das Maß definiert das Verhältnis der Summe der Streuung der Punkte zwischen den Clustern und der Streuung der Punkte innerhalb eines Clusters. Der Wert ist also höher, wenn die Cluster gut getrennt und homogen sind.

4 Datensatz

Als Datengrundlage für die Entwicklung einer Klassifikationspipeline zur Defekterkennung auf binären Waferkarten dient der von [44] zur Verfügung gestellte Satz von 38.000 Waferkarten aus SEM-Qualitätsmessungen. Eine Beobachtungseinheit ist eine binäre Waferkarte mit drei z-Werten – 0: Hintergrund, 1: Normal, 2: Fehler (Defekt) – in einer x-y-Ebene von 52x52 Pixeln. 38 unterschiedliche Defekte beschreiben einen Normaltyp (keine signifikanten Fehler), 8 Einzelfehlertypen und 29 aus Einzelfehlertypen gemischte Typen. Jeder Typ ist in gleichen Mengen vertreten. Die Einzelfehlertypen werden grafisch orientiert benannt mit Center (C), Donut (D), Edge-Loc (EL), Edge-Ring (ER), Loc (L), Near-Full (NF), Scratch (S), Random (R), die Mischtypen als Kombination und der Normaltyp mit Normal (N).

Die Autoren in [43] notieren zudem, dass der Datensatz mithilfe eines GAN [20] angereichert wurde, um die Messdaten aus den Qualitätsmessungen in Bezug auf die Häufigkeit des Auftretens der Fehler auszubalancieren, [6]. Als Ursache für das Entstehen von Defekten auf Silizium Trägerscheiben im Herstellungsprozess werden in [15] u. a. für C Probleme bei der mechanischen und

chemischen Planarisierung, EL als Folge von Dünnschichtablagerungen und ER und EL als Folge von Ätzproblemen genannt. [23] kategorisieren Defekte in zufällige, z.B. Partikel im Reinraum, systematische, auf den Prozess bezogene und steuerbare, und deren Kombination. [10] ermitteln die Ursache für spezifische Defekte, wie C durch Stromschwankungen, unregelmäßige Spülung oder Druckverluste, L durch Ventil- oder Pumpenstörungen, R durch minimale Kontamination der Prozessapparatur, und S durch Übergabefehler der Robotersteuerungen oder Transportfehler. [26] und [24] beschreiben darüber hinaus, wie Defektkombinationen in Einzeltypen mittels Methoden des maschinellen Lernens oder bildverarbeitende Techniken aufgelöst und damit neue Defekttypen oder deren Kombinationen erkannt werden können.

Die Kennzeichnung der 38 Defekttypen, ist mit 8 Bit OHL vorbereitet. Jede Bitkombination repräsentiert genau einen Typ bzw. Kategorie. Eine 8 Bit kodierte Zeile wird als Label für eine Kategorie beschrieben und jeder Waferkarte eindeutig zugeordnet. Im Beispiel heißt das u.a., dass Bitkombinationen für die Einfachtypen C mit 10000000 und S mit 00000010 oder für Zweifachtypen D-L mit 01001000, wie in Abb. 2, aus [2] mit Hintergrund *weiß*, Normal *hell*, Defekt *dunkel*.

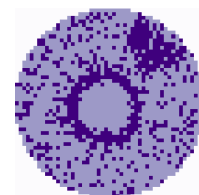


Abbildung 2. Beobachtungseinheit

Die Vergabe der Labels erfolgt zur Vorbereitung des Datensatzes durch manuelle menschliche Bewertung. In [45] wird anschaulich gezeigt, welche Probleme bei derart von Bewertungen auftreten und warum es erstrebenswert ist, die Analyse softwaregestützt zu automatisieren.

5 Ergebnisse

Die prototypische Entwicklung (Implementierungsdetails unter [2]) der Klassifikationspipeline als digitales Modell zur Erkennung von Wafer-Defekttypen ermittelt bei der datengetriebenen Analyse des Datensatzes schrittweise experimentelle Ergebnisse. So wird das Ziel erreicht, binäre Waferkarten anhand ihrer Illustrationen in normale und defekte, bekannte Typen und unbestimmte Typen, Ausreißer, zu unterteilen. Die Aufteilung des Datensatzes in Trainings-, Validierungs- und Testdaten erfolgt im Verhältnis 80 : 10 : 10%.

Schritt 1: Die Häufigkeitsanalyse in Abb. 3 aus [2] ermittelt das Verhältnis der drei z-Werte einer binären Waferkarte mit 0: Hintergrund, 1: Normal, 2: Fehler. Es wird ersichtlich, dass die Fehleranteile der Datensatzlabel 000: N, 001: R und 004: NF Defekttypen erheblich von den Anteilen der anderen abweichen. Für die darauf folgende Untersuchung auf Ausreißer werden daher die Stichproben aus N, R und NF ausgeschlossen. Die weiterführende Beurteilung in diesen Klassen endet damit aufgrund der hohen Defektpunktzahl. Zur Erkennung von Ausreißern innerhalb der Defekttypen werden nun die Defektpunkte ins Verhältnis zu ihren Normalpunkten gesetzt und die anteilige Verteilung in Kastendiagrammen (engl. boxplot) abgetragen. Sie sind ein probates Mittel, um über robuste Lage-

und Streuungsmaße Eigenschaften der Verteilung zu ermitteln. Außerhalb der Spannweite liegende Stichproben können so leicht lokalisiert werden. So sei bspw. genannt, dass im 2-fach-Defekttyp mit dem Label 096 für die Kombinationen [32, 64] für EL-D, die Ausprägung des D wesentlich dünner ausfällt.

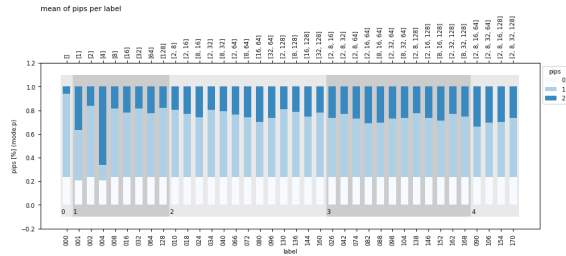


Abbildung 3. Häufigkeitsanalyse

Schritt 2: Die Güte des neuronalen Lernalers kann bereits während des Trainings gemessen werden. Dazu werden dem CNN nach obiger Auslegung nicht nur Trainings-, sondern auch Validierungsdaten präsentiert. Nach 25 ausgeführten Epochen stellen sich für die Verlustfunktion und Gütemaße entsprechende Werte ein, mit $loss_{train} = 0.0790$ und $loss_{valid} = 0.1900$ oder $acc_{train} = 0.9754$ und $acc_{valid} = 0.9577$.

Die Trainings- und Validierungsverläufe der Abb. 4 aus [2] der Verlustfunktion und Fläche unter der ROC-Kurve über 25 Epochen zeigen das nach einer kurzen Einstellphase angestrebte Ziel der Minimierung zu 0 bzw. Maximierung zu 1.

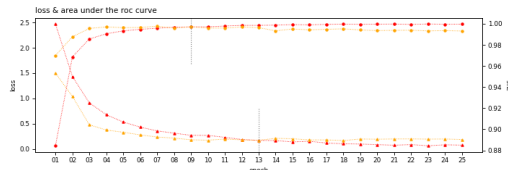


Abbildung 4. Verlustfunktion, Fläche unter der ROC

Es sind Kreuzungspunkte der Verläufe in Epoche 13 für die Verlustfunktion $loss$ und in der Epoche 9 für die Fläche unter der ROC-Kurve auc zu beobachten. Es wird vermutet, dass hier bereits das CNN adäquat auf den Trainingsdatensatz eingestellt ist und das weitere Training zu einer Über- evtl. auch Unteranpassung führt. Eine weiteres experimentelles Setup zur Verbesserung der Güte der Vorhersage scheint möglich. Die Auslegung des CNN sollte intern zu untersuchen sein ([2], wisa.ins.4).

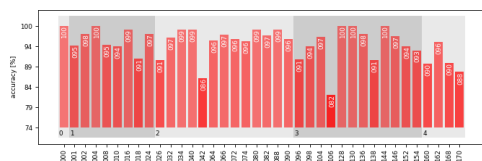


Abbildung 5. Diagonalwerte der Konfusionsmatrix

Bei der Vorhersage der Normal- und Defekttypen durch das trainierte CNN ist insbesondere die Genauigkeit der richtig getroffenen Aussagen von Interesse. Dem neuronalen Lerner werden Testdaten

präsentiert und mittels der diagonalen Werte der Konfusionsmatrix analysiert, Abb. 5 aus [2].

Das CNN erkennt nicht alle Defekttypen gleichermaßen gut. Insbesondere die Mischtypen 042 und 106 weisen auf Probleme hin, die in Verbindung mit unzureichender Fehlererkennung durch Defekttypüberlagerung auftreten. So wird bei der 042 als eine Kombination aus [2, 8, 32] (S-L-EL), S häufig durch L überlagert. Bei Typ 106 der Kombinationen [2, 8, 32, 64] (S-L-EL-D) verdecken diese sich ebenfalls gegenseitig. Die Fehlermatrix und eine anteilige Mengenausgabe für den Testdatensatz zeigt jedoch, dass lediglich 0.11% der erwarteten Einordnungen falsch klassiert werden. Der CNN-Klassifikator wird daher als gut, mit Potential zur Verbesserung, angenommen.

Schritt 3: Eine Visualisierung der Vorhersageergebnisse des CNN aus dem vorangegangenen Schritt ergibt die charakteristische zweidimensionale Abbildung 6 aus [2] des durch den t-SNE entwickelten Merkmalsraumes. Die einzelnen Defekttypen sind gut voneinander zu trennen und werden markant dargestellt. Gut zuerkennen sind auch Typübergänge (kreisförmige Markierung), bspw., wenn ein neuer Fehler hinzukommt. Es wird eine Divergenz nach [29] von 0.8 erreicht. Die Abbildung des hoch- auf den niedrig dimensional Raum wird daher als gut bewertet.



Abbildung 6. t-SNE

Schritt 4: Der in Schritt drei erzeugte Merkmalsraum wird nun noch einer klassenanteiligen Cluster-Analyse unterzogen, um mithilfe eines dichte-basierten Verfahrens nicht dichteverbundene Stichproben innerhalb der Defekttypen so zu erkennen. Wie zu erwarten, zeigen die DBSCAN nicht verketete Stichproben rein maschinell als Ausreißer an. DBSCAN liefert dabei eine Liste von Labels, welche mit -1 gelabelte Stichproben als Ausreißer klassieren. Die den DBSCAN bewertenden mittleren Maßzahlen aller Defekttypen ergeben sich für Silhouette (SC), Davis-Bouldin (DB) und Calinski-Harabasz (CH) zu $SC = 0.756$, $DB = 0.522$, $CH = 187.849$. Der SC, mit einem Wert gegen 1, DB, mit einem kleinen Betrag und CH, mit einem hohen Wert, weisen darauf hin, dass die Ausreißer gut von den eigentlichen Clustern getrennt sind und als solche anerkannt werden können. Es sei erwähnt, dass die Stichproben des Normaltyps eine Besonderheit aufweisen. Sie zerfallen in zwei räumlich getrennte Cluster, die ihrerseits jedoch keine wirklichen Ausreißer bilden und Auskunft zu zwei unterschiedlichen Arten von Normaltyp mit unterschiedlicher Defektpunktdichte geben.

6 Diskussion

Die Autoren in [43] erstellen ein DC¹⁹-Netz, dessen DC-Layer mit zweidimensionalen Offset-Matrizen aus geometrischen Transformationen arbeitet. Sie verschieben für die Weiterverarbeitung die

¹⁹DC, Abk. engl. Deformable Convolutional

Kernel-Masken durch Offsets in Richtung der interessanten Defekt-Punkte. Unter Verwendung des Datensatzes aus [44], bestehend aus den beschriebenen Einfach- und Mehrfach-Defekt-Typen, erreichen sie eine mittlere Genauigkeit von 0.932. Zudem vergleichen sie Netzimplementierungen von BCNN²⁰ und OH-CNN, welche weniger effektiv vorhersagen, da sie anstatt eines 38- nur ein 8-stelliges Encoding von OHL realisieren. Der Autor in [48] bietet einen Ansatz zur Fehlerkennung an, der sich im Bereich des QML²¹ bewegt. Der hybride Ansatz arbeitet auf durch Quantum-Schaltkreise extrahierten Merkmalen klassischer Fehlerbildeingaben des ebenfalls frei verfügbaren Testdatensatzes aus [14], beschrieben u.a. in [47]. Die lineare Verbindung, mit denen Neuronen üblicherweise über die Schichten miteinander verbunden sind, wird durch einen Quantenschaltkreis ersetzt. Damit kann den nicht linearen Eigenschaften der Merkmale besser Rechnung getragen werden. Das vorgestellte HCQDL²²-Modell bleibt allerdings hybrid, die effektive Nutzung bedarf weiterer Forschung. Es wird eine Genauigkeit für den Datensatz aus [47] mit 0.981 erreicht und mit Standard CNN- oder anderen bekannten Netz-Implementierungen verglichen, [38]. In [33] wird anschaulich ausgeführt, wie sich ein Ungleichgewicht der Klassen oder die Behandlung von Höhenverwerfungen an Fehlerpunkten (engl. warpage) auf die Genauigkeit der Vorhersagen auswirkt. Dabei stellen die Autoren pro Anreicherungsmethode auf dem Datensatz entsprechende Klassifizierer und Gütemaße vor und erreichen so u.a. eine mittlere Genauigkeit bei der Verwendung eines MLP-NN²³ von 0.95533, oder bei einer SVM²⁴ von 0.9428. Die Autoren in [37] stellen ein CNN vor, welches binäre Waferkarten mit einer 3-Kanal-Auflösung auf RGB²⁵-Daten der Dimension 224x224 aus verschiedenen Datensätzen verarbeitet und nutzen als nachbearbeitende Schritte t-SNE und HDBSCAN²⁶, um neue Defekttypen zu finden. Sie stellen einen Vergleich zur CNN-Modellierung und bereits etablierten Modellen auf und erreichen eine Trainingsgenauigkeit von 0.9541. Ergänzend sei erwähnt, dass die Autoren in [1] eine SVM mit einem RBF²⁷-Kernel implementiert wurde, um Wafer in gut und fehlerhaft mit einer Genauigkeit von 0.956 zu unterteilen. Sie stellen einen eigenen Datensatz vor, machen jedoch keine genauen Angaben zur Auflösung und Aufnahme.

Modell	Genauigkeit
DC-Netz	0.932
HCQDL	0.981
MLP-NN	0.955
SVM	0.943
HDBSCAN	0.954
SVM(RBF)	0.956
CNN	0.975

Tabelle 2. Tendenz der Genauigkeit im Vergleich

Die qualitative Bewertung des eigens entwickelten Modells für die Klassifikationspipeline zur Defekterkennung auf binären Waferkarten erfolgt durch Vorlage von Testdaten anhand von numerischen oder strukturellen Gütemaßen. Sie geben Auskunft, wie gut

das Modell in der Lage ist, unbekannte Testfehler-Waferkarten in die zuvor erlernten Klassen richtig einzuordnen. Ausreißer in den Testdaten werden sowohl in der Vor- als auch in der Nachklassifikation erfolgreich aussortiert. Der Anteil detektierter Ausreißer liegt bei etwa 0.11%. Das implementierte CNN erreicht im Test eine Genauigkeit von 0.9754. Die Fläche der Operationscharakteristik [17] wird mit 0.9997 angegeben. Unter Beachtung der Verwendung unterschiedlicher Datensatzausprägungen und Methoden zur Vorhersage werden gegenüberstellend zu obigen Ausführungen, die Vorhersagen des Netzes der hier entwickelten Teilmodelle der Klassifikationspipeline als hinreichend genau eingestuft und in der Tendenz, und aus der Tabelle 2 zu entnehmen, als gut bewertet.

Es wurde gezeigt, dass mithilfe der prototypisch entwickelten Klassifikationspipeline bekannte und unbekannte Wafer-Defekte erkannt werden können. Mithilfe eines geeigneten und aufbereiteten Trainings- und Validierungsdatsatzes und mittels einer prototypischen Implementierung werden die qualitative Bewertung von bekannten und unbekannten Fehlerbildern auf binären Waferkarten durch ein geeignetes Modell nachgewiesen. Der neuronale Lerner, der nur bereits bekannte Fehlerbilder zu erkennen vermag, wird durch vor- und nachgelagerte Methoden des maschinellen Lernens, beim Auffinden von Ausreißern und unbekannten Fehlerklassen zusätzlich unterstützt. Es wird darauf hingewiesen, dass der errechnete Verlust der Validierungsdaten im Vergleich zu den Trainingsdaten höher ist. Hier wird eine Unteranpassung (engl. underfitting) des CNN vermutet. Dieser Umstand ist evtl. lösbar, wenn über eine höhere Anzahl von Epochen trainiert, die Adaption der Lernrate geändert oder die Trainingsdatenmenge durch entsprechende Anreicherungsmechanismen vergrößert werden. In der Nachbearbeitung werden dimensionsreduzierende und bereichseinordnende Maßnahmen eingeführt, um weitere Ausreißer oder unbekannte Defekte erkennen zu können. Die Reduktion der Dimension der Ausgaben des Netzes auf zwei Komponenten erfolgt mithilfe eines nicht linearen Embedding-Algorithmus. Es entstehen sichtbare zweidimensionale Teilbereiche für die jeweiligen Defektklassen. Diese Bereiche werden einer weiteren Beurteilung durch einen Cluster-Algorithmus unterzogen, um so zusätzlich einzuordnende Stichproben in Form von Ausreißern zu finden, die sich nicht im Dichteverbund eines Defekttyps befinden. Die für die Teilmodelle erforderlichen Hyperparameter werden experimentell ermittelt und können bei Verwendung eines anderen Datensatzes variieren.

7 Ausblick

Die Ergebnisse der Modellkopplung, insbesondere die Analyse-schritte aus Vorverarbeitung und Nachbearbeitung, sollten zur Ermittlung neuer unvorteilhafter Prozesszustände oder Betriebsmittelparame-ter beitragen. Weiterführend kann untersucht werden, bei welcher Auftrittshäufigkeit neue spezifische Ausreißer als eine neue Klasse dem neuronalen Lerner zugeführt werden können oder wie eine teilautomatisierte Adaption einzurichten ist. Weitergehende Fragestellungen zur Plausibilität der Ergebnisse sollten erörtert und Grenzen in der Ergebnis-Ermittlung aufgezeigt werden. Inwieweit könnte demnach die prototypische Implementierung als digitaler Zwilling im Online-Processing zur Fehlerklassifikation einzusetzen sein. Wie gut wäre die Performance des vorgestellten Modells bei der Verwendung anderer frei verfügbarer und größerer Datensätze, bspw. mit [14]. Ein Analyse der Auslegung der Layer des CNN nach [2], wisa.ins.4 würde evtl. zu einer Verbesserung der Vorhersagequalität führen. Mittels Ansätzen der geometrischen

²⁰BCNN, Abk. engl. Basic CNN

²¹QML, Abk. engl. Quantum Machine Learning

²²HCQDL, Abk. engl. Hybrid Classical-Quantum Deep Learning

²³MLP-NN, Abk., engl. Multi Layer Perceptron Neural Network

²⁴SVM, Abk., engl. Support Vector Machine

²⁵RGB, Abk., Red Green Blue

²⁶HDBSCAN, Abk., engl. Hierarchical DBSCAN

²⁷RBF, Abk., engl. Radial Base Function

Anreicherung könnte nach [2], wisa.aug.4 die Menge der Daten pro Klasse gleichanteilig angehoben werden, um so eine größere und diversere Trainingsdatenmenge zu erhalten oder neue Fehlerkombinationen zu erhalten. Schlussendlich sind weitere Schrittfolgen aus den Teilmodellen je nach Anwendungsfall denkbar, z.B. die direkte Einzel-Typisierung und Ausreißer-Erkennung ohne statistische Vorverarbeitung oder eine Einzel- und Gruppen-Typisierung und Ausreißer-Erkennung nach Fehlertypanzahl mit statistischer Vorverarbeitung.

Literatur

- [1] Ramy Baly and Hazem Hajj. 2012. Wafer classification using support vector machines. *IEEE Transactions on Semiconductor Manufacturing* 25, 3 (2012), 373–383.
- [2] Norman Bauersfeld. 2023. *Wafer-to-wafer variations, Jupyter Quellen zu Model und Test*. Hochschule Harz. Retrieved 03.2023 from <https://github.com/nbauersfeld/wisa.git>
- [3] Christopher M Bishop and Nasser M Nasrabadi. 2006. *Pattern recognition and machine learning*. Vol. 4. Springer, New York.
- [4] Léon Bottou. 2012. *Stochastic Gradient Descent Tricks*. Springer, Berlin/Heidelberg, 421–436.
- [5] Günther Bourier. 2013. *Beschreibende Statistik: Praxisorientierte Einführung*. Springer, Berlin/Heidelberg.
- [6] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. 2018. A systematic study of the class imbalance problem in convolutional neural networks. *Neural networks* 106 (2018), 249–259.
- [7] Tadeusz Caliński and Jerzy Harabasz. 1974. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods* 3, 1 (1974), 1–27.
- [8] Young-Jin Cha, Wooram Choi, and Oral Büyükköztürk. 2017. Deep learning-based crack damage detection using convolutional neural networks. *Computer-Aided Civil and Infrastructure Engineering* 32, 5 (2017), 361–378.
- [9] Sejeun Cheon, Hankang Lee, Chang Ouk Kim, and Seok Hyung Lee. 2019. Convolutional neural network for wafer surface defect classification and the detection of unknown defect class. *IEEE Transactions on Semiconductor Manufacturing* 32, 2 (2019), 163–170.
- [10] Jong-Chih Chien, Ming-Tao Wu, and Jiann-Der Lee. 2020. Inspection and classification of semiconductor wafer surface defects using CNN deep learning networks. *Applied Sciences* 10, 15 (2020), 5340.
- [11] Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement* 20, 1 (1960), 37–46.
- [12] Sean P Cunningham and Scott Mackinnon. 1998. Statistical methods for visual defect metrology. *IEEE Transactions on Semiconductor Manufacturing* 11, 1 (1998), 48–53.
- [13] David L Davies and Donald W Bouldin. 1979. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence PAMI-1*, 2 (1979), 224–227.
- [14] Yi Wang, Dong Ni. 2022. *Synthesized Multi-GFA Multi-bin Wafer Bin Map Dataset*. IEEE. Retrieved 04.2022 from <https://dx.doi.org/10.21227/s1je-w708>
- [15] Rajesh Doss, Jayabrabu Ramakrishnan, S Kavitha, S Ramkumar, G Charlyn Pushpa Latha, and Kiran Ramaswamy. 2022. Classification of Silicon (Si) Wafer Material Defects in Semiconductor Choosers using a Deep Learning ShuffleNet-v2-CNN Model. *Advances in Materials Science & Engineering* 2022 (2022), 1–12.
- [16] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (Portland, Oregon) (KDD '96)*. AAAI Press, Portland Oregon, 226–231.
- [17] Tom Fawcett. 2004. ROC graphs: Notes and practical considerations for researchers. *Machine learning* 31, 1 (2004), 1–38.
- [18] Hongbin Gao, Ya Zhang, Wenkai Lv, Jiawei Yin, Tehreem Qasim, and Dongyun Wang. 2022. A deep convolutional generative adversarial networks-based method for defect detection in small sample industrial parts images. *Applied Sciences* 12, 13 (2022), 6569.
- [19] Aurélien Géron. 2020. *Praxiseinstieg Machine Learning mit Scikit-Learn, Keras und TensorFlow: Konzepte, Tools und Techniken für intelligente Systeme. Aktuell zu TensorFlow 2*. O'Reilly, Heidelberg.
- [20] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144.
- [21] Wang Hao, Wang Yizhou, Lou Yaqin, and Song Zhili. 2020. The role of activation function in CNN. In *2020 2nd International Conference on Information Technology and Computer Application (ITCA)*. IEEE, IEEE, Guangzhou, China, 429–432.
- [22] Haibo He and Edwardo A Garcia. 2009. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering* 21, 9 (2009), 1263–1284.
- [23] Chia-Yu Hsu, Wei-Ju Chen, and Ju-Chien Chien. 2020. Similarity matching of wafer bin maps for manufacturing intelligence to empower industry 3.5 for semiconductor manufacturing. *Computers & Industrial Engineering* 142 (2020), 106358.
- [24] Chenn-Jung Huang, Chi-Feng Wu, and Chua-Chin Wang. 2002. Image processing techniques for wafer defect cluster identification. *IEEE Design & Test of Computers* 19, 2 (2002), 44–48.
- [25] Yong Sung Ji and Jee-Hyong Lee. 2020. Using GAN to improve CNN performance of wafer map defect type classification: Yield enhancement. In *2020 31st Annual SEMI Advanced Semiconductor Manufacturing Conference*. IEEE, Korea, 1–6.
- [26] Cheng Hao Jin, Hyuk Jun Na, Minghao Piao, Gouchol Pok, and Keun Ho Ryu. 2019. A novel DBSCAN-based defect pattern detection and classification framework for wafer bin map. *IEEE Transactions on Semiconductor Manufacturing* 32, 3 (2019), 286–292.
- [27] M Kannan. 2018. *Scanning electron microscopy: Principle, components and applications*. Springer, Berlin/Heidelberg, 81–92.
- [28] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2017. Imagenet classification with deep convolutional neural networks. *Commun. ACM* 60, 6 (2017), 84–90.
- [29] Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics* 22, 1 (1951), 79–86.
- [30] Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. 2011. Stacked convolutional auto-encoders for hierarchical feature extraction. In *Artificial Neural Networks and Machine Learning—ICANN 2011: 21st International Conference on Artificial Neural Networks, Espoo, Finland, June 14-17, 2011, Proceedings, Part I* 21. Springer, Berlin/Heidelberg, 52–59.
- [31] J D Meindl. 1987. Chips für künftige Computergenerationen. *Spektrum der Wissenschaft* 1987, 12 (1987), 64–71.
- [32] James Moyné, Enrique Del Castillo, and Arnon M Hurwitz. 2018. *Run-to-run control in semiconductor manufacturing*. CRC press, Boca Raton, Florida.
- [33] You-Jin Park, Rong Pan, and Douglas C Montgomery. 2023. A novel hybrid resampling for semiconductor wafer defect bin classification. *Quality and Reliability Engineering International* 39, 1 (2023), 67–80.
- [34] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research* 12 (2011), 2825–2830.
- [35] Irene Rößler and Albrecht Ungerer. 2008. *Statistik für Wirtschaftswissenschaftler*. Springer, Berlin/Heidelberg.
- [36] Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics* 20 (1987), 53–65.
- [37] Francesco Rundo, Salvatore Coffa, Michele Calabretta, Riccardo Emanuele Sarpietro, Angelo Messina, Carmelo Pino, Simone Palazzo, and Concetto Spampinato. 2022. Hybrid Deep Learning Pipeline for Advanced Electrical Wafer Defect Maps Assessment. In *AEIT*. IEEE, Rome, 1–6.
- [38] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE, Salt Lake City, 4510–4520.
- [39] Patrice Y Simard, David Steinkraus, John C Platt, et al. 2003. Best practices for convolutional neural networks applied to visual document analysis. In *Icdar*, Vol. 3. Microsoft, Redmond, 0–0.
- [40] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.
- [41] Ganesha Udupa, BKA Ngoi, HC Freddy Goh, and MN Yusoff. 2003. Defect detection in unpolished Si wafers by digital shearography. *Measurement Science and Technology* 15, 1 (2003), 35.
- [42] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008), 2579–2605.
- [43] Junliang Wang, Chuqiao Xu, Zhengliang Yang, Jie Zhang, and Xiaoli Li. 2020. Deformable convolutional networks for efficient mixed-type wafer defect pattern recognition. *IEEE Transactions on Semiconductor Manufacturing* 33, 4 (2020), 587–596.
- [44] J. X. Wang. 2022. *Mixed-type Wafer Defect Datasets*. Donghua University. Retrieved 10.2022 from <https://www.kaggle.com/datasets/co1d7era/mixedtype-wafer-defect-datasets>
- [45] M-JJ Wang and Chung-Lun Huang. 2004. Evaluating the eye fatigue problem in wafer inspection. *IEEE Transactions on Semiconductor Manufacturing* 17, 3 (2004), 444–447.
- [46] Tilo Wendler and Sören Gröttrup. 2021. Case Study: Fault Detection in Semiconductor Manufacturing Process. In *Data Mining with SPSS Modeler: Theory, Exercises and Solutions*. Springer, Berlin/Heidelberg, 1193–1248.
- [47] Ming-Ju Wu, Jyh-Shing R Jang, and Jui-Long Chen. 2014. Wafer map failure pattern recognition and similarity ranking for large-scale data sets. *IEEE Transactions on Semiconductor Manufacturing* 28, 1 (2014), 1–12.
- [48] Yuan-Fu Yang and Min Sun. 2022. Semiconductor defect detection by hybrid classical-quantum deep learning. In *CVPR*. IEEE, New Orleans, Louisiana, 2323–2332.
- [49] Qing Zhang, Yuhang Zhang, Jizuo Li, and Yongfu Li. 2022. WDP-BNN: Efficient wafer defect pattern classification via binarized neural network. *Integration* 85 (2022), 76–86.