Grundprojekt Sommer 2019
Master Informatik

Simulationsbasierter Vergleich von Methoden zur Anomaly Detection im Netzwerk des Fahrzeugs

Wilhelm Schumacher

Eingereicht am XX.XX.2019

Zusammenfassung Mit dem immer weiter voranschreitenden Ausbau der Kommunikation im internen Netzwerkes des Autos, wie zum Beispiel der Einführung von Echtzeit-Ethernet mit TSN im zentralen Backbone des Netzes und auf lange Sicht die Entwicklung hin zu einem flachen Ethernet-Netzwerk, steigt auch gleichzeitig der Bedarf an IT Security Konzepten im Auto. Zusätzlich erfolgt im Moment eine Öffnung des internen Netzes hinzu vermehrter Kommunikation mit der Außenwelt. Ein Security Konzept, das in diesem Kontext im Moment wachsende Aufmerksamkeit erhält, ist die Entdeckung von Angriffen durch Anomalieerkennung. Unterschiedliche Verfahren zur Anomalieerkennung werden bereits in anderen Domänen wie zum Beispiel bei der Erkennung von Kreditkartenbetrug, im medizinischen Bereich, für Fehlererkennungen im Raumschiff oder klassisch für Intrusion Detection Systeme, die Angriffe auf Netzwerke entdecken sollen, erfolgreich eingesetzt. Das langfristige Ziel ist es Herauszufinden, wie praktikabel der Einsatz von Anomalieerkennung im Fahrzeug ist und welche der bekannten Algorithmen besonders gut für die Erkennung von Anomalien innerhalb der Automotive Security Domäne geeignet sind. Diese Ausarbeitung gibt erstmal einen geordneten Überblick zu den Methoden der Anomalieerkennung. Anschließend ist das Ziel dieser Ausarbeitung innerhalb des Simulations-Framework OMNeT++ (Objective Modular Network Testbed in C++) in einem beispielhaften modellierten Fahrzeug-Netzwerk Basisalgorithmen zur Anomalieerkennung aus unterschiedlichen Kategorien miteinander zu vergleichen. Dafür werden nach einer kurzen Lernphase mit normalen Daten verschiedene Angriffsszenarien durchgeführt werden, die anschließend durch die Algorithmen korrekt erkannt werden müssen.

Schlüsselwörter Automotive Security · Anomaly Detection · Fahrzeugnetzwerk · Anomalieerkennung · Vergleich von Anomaly Detection Methoden · OMNeT++

Wilhelm Schumacher
E-Mail: Wilhelm.Schumacher@haw-hamburg.com
Department Informatik
Fakultät Technik und Informatik
Hochschule für Angewandte Wissenschaften Hamburg

Inhaltsverzeichnis

1 Einleitung

Moderene Fahrzeugnetzwerke bestehen aus einer Vielzahl verteilter Steuereinheiten (ECUs) [vgl. SNA+13, Seite 2-3], die über verschiedene Kommunikationsmedien systemweit viele Informationen über Betriebszustände und weitere relevante Daten miteinander austauschen. Typischerweise enthält ein modernes Auto über 70 verschiedene ECUs. Zur Verbindung der ECUs werden als Kommunikationsmedien entweder verschiedene Systembusse wie CAN, LIN, MOST, FlexRay oder vor allem in neueren Fahrzeugen Ethernet-Netzwerke [HMVVDK13], die Systembusse langfristig ersetzen sollen, eingesetzt. Auch die Angriffsoberfläche (attack surface) des Fahrzeuges hat sich durch neue Schnittstellen, wie einer Internetverbindung zur Außenwelt, Car-To-X, OBD-II, Bluetooth usw., mit der Zeit immer mehr erweitert [vgl. CMK+11, Seite 2-4]. Diese Interfaces bieten das Potential für die Umsetzung einer Vielzahl von neuen Funktionen, mit Updates over the Air kann zum Beispiel das Kartenmaterial des autonomen Fahrzeuges aktualisiert werden, mit Car-To-X Kommunikation die Motorsteuerung an die aktuelle Verkehrslage angepasst werden oder mit Hilfe der Cloud der nächste Ladezyklus des Elektroautos inklusive Reservierung und Einbeziehung des Kalenders besser geplant werden.

Jedoch führt die Öffnung des internen Kommunikationsnetzes nach Außen auch zu einer erhöhten Verwundbarkeit der Informationssicherheit (Integrität, Vertraulichkeit, Verfügbarkeit) im internen Netzwerk des Fahrzeuges [vgl. MA11, Seite 1]. Denn durch die Einbindung der neuen Interfaces wird das Auto zu einem attraktiven Angriffsziel für Hacker. Hacker könnten potentielle Schwachstellen in einem der externen Interfaces ausnutzen, um böswillige Pakete in das Netzwerk einzuschleusen, die den normalen Betriebsablauf des Fahrzeuges stören sollen. Zum Beispiel sind die Steuergeräte (ECUs) angreifbar und können nach Kompromittierung genutzt werden, um die gesamte Kommunikation zu manipulieren. Deswegen sind IT Security Konzepte zum Schutz des Fahrzeuges notwendig. Ein Security Konzept, das in diesem Kontext im Moment wachsende Aufmerksamkeit erhält, ist die Entdeckung von Angriffen durch Anomalieerkennung. Unterschiedliche Verfahren zur Anomalieerkennung werden bereits in anderen Domänen wie zum Beispiel bei der Erkennung von Kreditkartenbetrug, im medizinischen Bereich, für Fehlererkennungen im Raumschiff oder klassisch für Intrusion Detection Systeme, die Angriffe auf Netzwerke entdecken sollen, erfolgreich eingesetzt. Einige Algorithmen wurden dabei speziell für diese bestimmten Anwendungsbereiche entwickelt, abhängig von Art der Daten, der Verfügbarkeit von gelabelten Trainingsdaten, den Typen von Anomalien und dem gewünschten Outputformat [vgl. CBK09, Seite 6-7].

Dieses Grundprojekt soll einen ersten kleinen Beitrag zu dem langfristigen Ziel Herauszufinden, wie praktikabel der Einsatz von Anomalieerkennung im Fahrzeug ist und welche der bekannten Algorithmen besonders gut für die Erkennung von Anomalien innerhalb der Automotive Security Domäne geeignet sind, leisten. Dazu gibt diese Ausarbeitung erstmal einen geordneten Überblick zu Methoden der Anomaly Detsection aus der Literatur und zeigt auf wie diese in Kategorien unterteilt werden können. Anschließend ist das Ziel dieser Ausarbeitung innerhalb des Simulations-

Framework OMNeT++ (Objective Modular Network Testbed in C++) in einem beispielhaften modellierten Fahrzeug-Netzwerk Basisalgorithmen zur Anomalieerkennung aus unterschiedlichen Kategorien miteinander zu vergleichen. Die Simulation basiert auf Daten aus einer echten Kommunikationsmatrix und bildet korrekt ab welche Komponente mit welcher anderen Komponente kommuniziert. Anhand dieser Daten können die Algorithmen erstmal in einer Lernphase das normale Verhalten des Systems lernen. Darauffolgend werden verschiedene Angriffsszenarien durchgeführt, an denen die Algorithmen dann evaluiert werden, indem betrachtet wird, ob die Angriffe korrekt erkannt wurden. Die Arbeit findet innerhalb des SecVI Forschungsprojekt [siehe Sec] statt. Das SecVI Projekt wird in Verbindung mit Industriepartnern durchgeführt und vom Federal Ministry of Education und Research gefördert und hat als Ziel eine sichere Netzwerkarchitektur für das Auto zu entwickeln.

Die folgende Ausarbeitung ist so strukturiert, dass zuerst das Kapitel **Überblick Anomaly Detetction** relevante Grundlagen zum Thema Anomaly Detection beschreibt. Anschließend betrachtet das Kapitel **Methoden der Anomalieerkennung** die Zuordnung von Verfahren der Anomalieerkennung in Kategorien und erklärt die wichtigsten Charakteristika der einzelnen Kategorien. Das Kapitel **Aufbau der Simulationsumgebung** gibt einen Überblick zu der Funktionsweise der verwendeten Algorithmen, Metriken, Parametern und den Angriffsszenarien. Daraufhin stellt das Kapitel **Durchführung und Ergebnisse des Vergleiches** die Ergebnisse der Durchführung dar, die dann im folgenden Kapitel **Evaluierung der Ergebnisse** genauer betrachtet und interpretiert werden. Das letzte Kapitel **Zusammenfassung** schließt die Ausarbeitung dann mit einer kurzen Zusammenfassung der wichtigsten Punkte der Ausarbeitung und einem Ausblick hinsichtlich des Hauptprojektes ab.

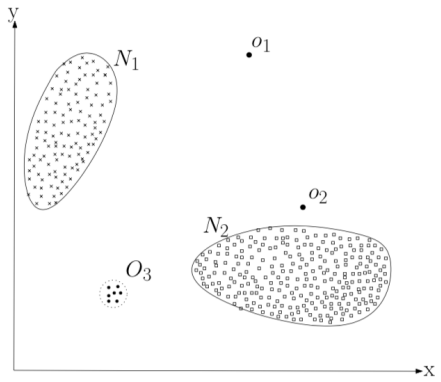
2 Überblick Anomaly Detection

Das Kapitel Überblick Anomaly Detetction erklärt die Grundlagen zum Thema Anomaly Detection, die für das Verständnis der folgenden Kapitel benötigt werden. Dazu erläutert das Unterkapitel 2.1 **Ziele der Anomaly Detection** zuerst einmal die Begriffe Anomalie und Anomalie Detection und fasst deren Funktionsweise, Anwendungsbereiche und Ziele kurz zusammen. Anschließend folgen die Kapitel 2.2 **Input Daten**, 2.3 **Typen von Anomalien** und 2.4 **Lernmethoden** mit weiteren wichtigen Aspekten zur Anomaly Detection.

2.1 Ziele der Anomaly Detection

Die Anomaly Detection hat als Aufgabe unregelmäßige Muster innerhalb von normalen Daten festzustellen. Diese Unregelmäßigkeiten entsprechen dabei nicht dem normalen Verhalten des Systems, treten seltener auf und unterscheiden sich signifikant vom Rest der Daten. Sie werden hauptsächlich als **Anomalien** oder **Outlier** (Ausreißer) [CBK09] bezeichnet. Anomaly Detection wird bereits in einer Vielzahl von unterschiedlichen Anwendungsbereichen eingesetzt. Zum Beispiel bei dem Erkennen von

Kreditkartenbetrug, zum Entdecken von Tumoren in MRT-Bilder (Magnetresonanz-tomographie), im Cyber Security Bereich, dem Erkennen von Cheatern in Videospielen [Pat] oder für die Fehlererkennung bei sicherheitskritischen Systemen [FYM05]. In



Quelle: http://doi.acm.org/10.1145/1541880.15418822 [vgl. CBK09, Seite 2]

Abb. 1 Abbildung 1 stellt die normalen Datenbereiche N1 und N2 den anomalen Beobachtungen O1,O2 und O3 gegenüber.

vielen dieser Anwendungsbereiche spielt die Identifikation von Anomalien eine große Rolle, denn sie ermöglicht es kritische Situationen zu erkennen und daraufhin die notwendigen Gegenmaßnahmen zu veranlassen. Zum Beispiel können ungewöhnliche Nachrichtenmuster in einem Computernetzwerk darauf hinweisen, dass von einem gehackten Computer aus, sensitive Daten in das Netzwerk an einen unautorisierten Host geschickt werden oder anomales Verhalten in einem Sensor eines Raumschiffes könnte auf Fehler in einer Komponente hinweisen.

Ein konkretes Beispiel für eine Anomalie ist in Abbildung 1 dargestellt. Hier können innerhalb eines zweidimensionalen Datensatz sowohl normale Bereiche (N_1 und N_2) als auch Anomalien (O_1, O_2 und O_3), die weiter von den restlichen Daten entfernt liegen, erkannt werden. Häufig ist es aber schwierig präzise Grenzen zwischen den normalen und anomalen Daten, wie in Abbildung 1, zu definieren und stellt eine der großen Herausforderungen der Anomaly Detection dar. Zum Beispiel können Anomalien, die dicht an der Grenze eines normalen Datenbereiches, nur sehr schwierig als Anomalie oder normaler Datensatz eingeordnet werden.

2.2 Input Daten

Abhängig vom Anwendungsbereich kann sich die Art der Input Daten deutlich voneinander unterscheiden. Im Allgemeinen sind die Input Daten erstmal eine Sammlung von Daten (bezeichnet als Objekte, Vektoren, Beobachtungen, Punkte etc.), die wiederum aus einer Menge von Attributen (bezeichnet als Variablen, Felder, Feature etc.) bestehen

[vgl. CBK09, Seite 6-7]. Ein Attribut stammt dabei aus einer von drei Kategorien **binär**, **kategorisch** oder **kontinuierlich**. Wenn die Daten nur aus einem Attribut bestehen, werden sie als **Univariate** bezeichnet und wenn die Daten aus mehreren Attributen bestehen als **Multivariate**. Je nachdem welche Arten von Attributen oder Kombinationen von Attributen innerhalb der Daten vorhanden sind, können andere Algorithmen angewendet werden oder die Daten müssen entsprechend angepasst werden. Zum Beispiel benötigen einige Algorithmen (Nearest Neighbor) ein Verfahren zur Distanzbestimmung, das für binäre Attribute anders funktioniert als für kategorische Attribute.

2.3 Typen von Anomalien

Eine Anomalie kann in eine von drei Kategorien [Gmb18] eingeordnet werden. Die erste und einfachste Kategorie wird als **Punktanomalie** bezeichnet. Bei einer Punktanomalie wird ein einziger Datenpunkt in Anbetracht zu den restlichen Daten als Anomalie eingestuft. Ein Beispiel aus der realen Welt für eine Punktanomalie wäre zum Beispiel die Höhe des ausgegebenen Betrages bei Bezahlung mit Kreditkarte. Wenn über einen längeren Zeitraum nur sehr kleine Beträge bezahlt wurden und dann ein im Vergleich zu diesen Beträgen sehr großer Betrag auftritt, handelt es sich um eine Punktanomalie. In der zweiten Kategorie **kollektive Anomalien** verhält sich eine Reihe von Datenpunkten abweichend vom Rest der Daten. Dabei sind die individuellen Datenpunkte aus der kollektiven Anomalie nicht unbedingt anormal, sondern nur deren gemeinsames Auftreten. Ein Beispiel für eine kollektive Anomalie ist das Signal eines Elektrokardiogramms [vgl. CBK09, Seite 9] . Wenn das Signal über eine längeren Zeitraum stabil bleibt und keine steilen Anstiege enthält wie in den anderen Intervallen, handelt es sich um eine kollektive Anomalie. Die dritte Kategorie von Anomalien sind die **kontextuellen** Anomalien. Bei einer kontextuellen Anomalie wird ein Wert nur in einem spezifischen Kontext als Anomalie betrachtet. Zum Beispiel könnte eine kontextuelle Anomalien in Temperatur-Zeitreihen auftreten. Ein sehr niedriger Temperaturwert im Winter wäre nicht ungewöhnlich, aber im Sommer wäre es eine kontextuelle Anomalie.

2.4 Lernmethoden

Bei der Implementierung von Anomalieerkennungen wird zwischen den drei Methoden überwachtes Lernen, semiüberwachtes Lernen und unüberwachtes Lernen unterschieden [vgl. BBK14, Seite 7]. Welche Methode angewendet werden kann, hängt hautsächlich davon ab, ob gelabelte Trainingsdaten oder nur ungelabelte Trainingsdaten zur Verfügung stehen. Bei gelabelten Trainingsdaten sind die Datenpunkte entweder als normal oder anormal markiert. Häufig ist es aber sehr schwierig und aufwendig, genug korrekt gelabelte Trainingsdaten zu erhalten, weil diese oft manuell von menschlichen Experten erarbeitet werden müssen. Beim überwachten Lernen wird vorausgesetzt das Datenpunkte sowohl für die normale Klasse als auch für die anormale Klasse verfügbar sind. Auf Basis dieser Eingabedaten soll das System dann ein Model erstellen, dass es

ermöglicht möglichst zielsicher vorauszusagen kann, ob ein neuer Datenpunkt eine Anomalie darstellt oder nicht. Dagegen wird beim semiüberwachtes Lernen nur auf Grundlage von Daten aus der normalen Klasse bestimmt, ob eine neue Dateninstanz anormal ist oder nicht. Ein Vorteil des semiüberwachten Lernens ist die bessere Anwendbarkeit, weil keine Daten für die Klasse der Anomalien benötigt werden. Für das unüberwachte Lernen werden überhaupt keine gelabelten Trainingsdaten benötigt. Es wird davon ausgegangen das normale Dateninstanzen in den Testdaten deutlich häufiger auftreten als Anomalien. Anomalien können dann dadurch identifiziert, indem Datenpunkte gefunden werden, die von den am häufigsten auftretenden Mustern abweichen. Ein großer Nachteil dieser Technik ist, dass wenn die Annahme nicht stimmt, das System sehr viel falsch positive Rückmeldungen bekommt.

3 Methoden der Anomalieerkennung

Das Kapitel Methoden der Anomalieerkennung blickt auf Methoden, die zur Anomalieerkennung verwendet werden und ordnet diese in Kategorien ein. Dazu betrachtet es verschiedene Ansätze zur Kategorisierung aus der Literatur. Es erläutert für die Kategorien Classification, Statistisch, Clustering und Nearest Neighbor die grundlegende Funktionsweise und nennt Beispiele für Algorithmen aus dieser Kategorie. Weiterhin betrachtet das Kapitel die Vor- und Nachteile der einzelnen Kategorien und gibt einen Ausblick auf weitere Algorithmen, die zur Anomalieerkennung benutzt werden können.

3.1 Klassen von Algorithmen

Verfahren zur Anomalieerkennung klar voneinander abzugrenzen ist ziemlich schwierig. Aus diesem Grund findet man auch verschiedene Ansätze von Kategorisierungen in der Literatur wieder. Drei Beispiele für verschiedenen Kategorien in die die Algorithmen eingeordnet werden können, sind in der Tabelle 1 dargestellt. In der Tabelle werden die Kategorien aus den Arbeiten von [BBK14], [CBK09] und [GTDVMFV09] einander gegenübergestellt. Gemeinsam ist bei allen drei Ausarbeitungen, dass jeweils eine Kategorie für die statistischen Verfahren existiert. Unterschiede finden sich vor allem in der Kategorie Clustering. Während bei [GTDVMFV09] Clustering-Verfahren zu der Kategorie Machine Learning zählen, haben [BBK14] und [CBK09] diese Art von Algorithmen in eine separate Kategorie eingeordnet. Zusätzlich werden bei [CBK09] aus den Clustering-Verfahren nochmal die Nearest Neighbor Based Verfahren in eine extra Kategorie abgetrennt. Die Trennung basiert auf der Annahme, dass Clustering-Verfahren den Fokus auf die Bildung des Clusters und die anschließende Einordnung legen. Im Gegensatz dazu fokussieren sich die Nearest Neighbor Based Verfahren auf die lokale Nachbarschaft der Dateninstanzen. Eine weitere Kategorie die in allen drei Arbeiten zu finden ist, sind die Klassifikationsalgorithmen (Machine Learning bei [GTDVMFV09]). Ergänzend dazu sind noch weitere Kategorien wie Combination Learner, Spectral, Information Theoretic und Knowledge Based vorhanden, die speziellere Verfahren zusammenfassen und einzig in jeweils einer Ausarbeitung

als Kategorie vorkommen.

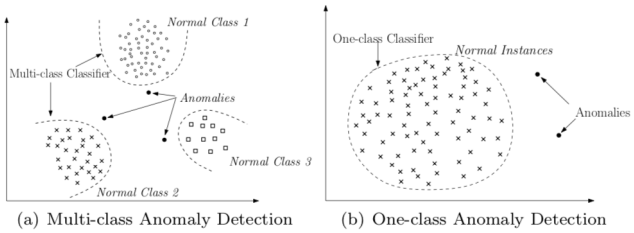
Bhuyan et al.: Network Anomaly Detection: Methods, Systems and Tools	Chandola et al.: Anomaly Detection : A Survey	Anomaly-based network intrusion detection: Techniques, systems and challenges
1. Statistical	1. Classfication Based	1. Statistical
2. Classification Based	2. Nearest Neighbor Based	2. Knowledge Based
3. Clustering and Outlier Based	3. Clustering Based	3. Machine Learning Based
4. Soft Computing	4. Statistical	-
5. Knowledge Based	5. Information Theoretic	-
6. Combination Learners	6. Spectral	-

Tabelle 1 zeigt beispielhaft Kategorien in die Algorithmen/Methoden zur Anomalieerkennung eingeordnet werden können.

3.2 Classification Based Verfahren

Die erste Kategorie von Verfahren stellen die Classification Based Verfahren dar. Classification Based Verfahren versuchen neu auftretende Dateninstanzen anhand eines zuvor gelernten Models in verschiedene Klassen einzuordnen [vgl. BBK14, Seite 12]. Dies geschieht in 2 Phasen. Zuerst erfolgt eine Trainingsphase in der gelabelte Trainingsdaten benutzt werden, um das Model zu lernen. Anschließend wird in einer Testpahase durch einen Klassifikator mit Hilfe des Models entschieden, ob es sich um eine normale oder anormale Instanz handelt. Dabei wird zwischen Verfahren mit nur einer normalen Klasse (one-class) und Multi-Klassen (multi-class) unterschieden. Bei Multi-Klassen Verfahren wird angenommen, dass er mehrere normale Klassen gibt, in welche die Dateninstanzen eingeordnet werden können. Wenn der Klassifikator der Dateninstanz keine der normalen Klassen sicher zuordnen kann, wird diese Dateninstanz als anormal eingestuft. In Abbildung 2 ist die Unterscheidung zwischen den beiden Verfahren dargestellt und es wird gezeigt was in dem jeweiligen Kontext eine Anomalie ist.

Beispiele für Klassifikationsalgorithmen sind verschiedene Varianten von neuronalen Netzen, bayessche Netze, Support Vector Machines oder regelbasierte Systeme. Bei Klassifikationsalgorithmen besitzt die Trainingsphase abhängig vom konkreten Algorithmus zwar eine hohe Rechenkomplexität, dafür ist die Testphase zum Einstufen neuer Beobachtungen wiederum sehr schnell [vgl. CBK09, Seite 21]. Dies stellt einen großen Vorteil der Klassifikationsalgorithmen dar. Ein weiterer Vorteil ist, dass zum Beispiel die multi-class Verfahren es erlauben zwischen verschiedenen normalen Klassen zu unterscheiden. Nachteile von Klassifikationsalgorithmen sind, dass viele Verfahren sehr abhängig von korrekt gelabelten Trainingsdaten sind und als Output nur angeben, ob eine Dateninstanz zu einer Klasse gehört anstatt einen Score anzugeben, der verdeutlicht wie wahrscheinlich es sich um eine anormale Instanz handelt.



(a) Multi-class Anomaly Detection (b) One-class Anomaly Detection

Quelle: http://doi.acm.org/10.1145/1541880.15418822 [vgl. CBK09, Seite 21]

Abb. 2 Abbildung 2 verdeutlicht die Funktionsweise von one-class Classification und multi-class Classification und hebt hervor, wo die Anomalien jeweils zu finden sind.

3.3 Nearest Neighbor Based Verfahren

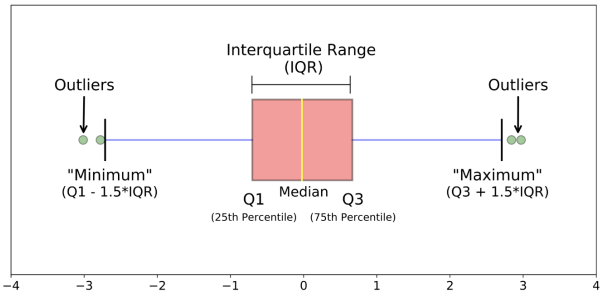
Nearest Neighbor Based Verfahren [vgl. CBK09, Seite 28] benutzen die Distanz bzw. die Ähnlichkeit (bei kategorischen Attributen) zwischen zwei Datenpunkten als Grundlage zur Anomalieerkennung. Abhängig von den Distanzen/Ähnlichkeiten ihrer Attribute liegen Datenpunkte verschieden weit auseinander. Mit der Annahme, dass normale Daten gesammelt in kurzer Distanz zueinander liegen und anormale Daten weit von ihren Nachbarn entfernt liegen, sollen Anomalien identifiziert werden. Welche Verfahren dabei zur Distanzbestimmung verwendet werden können, hängt von der Art der Attribute der Daten ab und ob es sich um Univariate oder Multivariate (siehe 2.2) handelt. Zum Beispiel ist ein einfaches Verfahren zur Distanzbestimmung der Euklidische Abstand. Der Euklidische Abstand kann aber nur auf kontinuierliche Daten angewendet werden. Daher müssen bei kategorischen Datentypen oder gemischten Attributtypen andere Verfahren oder Kombinationen von Verfahren benutzt werden. Die Funktionsweise von Nearest Neighbor Based Verfahren kann in die zwei Kategorien **kth Nearest Neighbor** und **Relative Density** unterteilt werden. Bei kth Nearest Neighbor Verfahren wird für eine Dateninstanz deren Distanz zu den nächsten k-Nachbarn als Anomalie-Score betrachtet. Anhand eines Schwellenwertes (threshold) wird die Dateninstanz dann als normal oder anormal eingestuft. Diese Basisvariante wurde von vielen Forschern noch erweitert, indem zum Beispiel nur die Nachbarn einer Dateninstanz, die nicht weiter als Distanz x entfernt liegen, gezählt werden, um zu Ermitteln bei welchen Daten es sich um Anomalien handelt [KNT00]. Bei Relative Density Verfahren wird eine Anomalie an der Dichte der Nachbarschaft erkannt. Es wird davon ausgegangen, dass Anomalien in Bereichen mit geringer Dichte an Nachbarn liegen, während normale Daten eine hohe Nachbarschaftsdichte besitzen. Der Abstand einer Dateninstanz zu seinem k entferntesten Nachbar stellt eine Schätzung der Inversen der Dichte im Datensatz dar. Wenn Datensätze mit variierenden Dichten vorliegen, wird es notwendig auch die Lokalität des Datensatzes in Relation zu seinen Nachbarn zu berücksichtigen. Ein Beispiel für ein Verfahren, das eine lokale Dichte anstatt einer globalen Dichte verwendet, ist der Local Outlier Factor [KKSZ09].

Ein Vorteil von Nearest Neighbor Based Verfahren ist, dass sie von Natur aus unsupervised sind und keine direkten Annahmen über die Verteilung der Daten treffen [vgl. CBK09, Seite 28]. Außerdem sind Nearest Neighbor Based Algorithmen gut anpassbar in Bezug auf verschiedene Datentypen, denn es wird nur ein anderes Verfahren zur Distanzbestimmung benötigt und Nearest Neighbor Based Verfahren können auch im semi-supervised Modus genutzt werden. Nachteile dieser Verfahren sind, dass normale Ausreißer ohne direkte Nachbaren als Anomalien eingeordnet werden könnten und Anomalien, die in Mitten von normalen Daten liegen, als normal eingestuft werden. Auch die Komplexität der Verfahren ist in der Testphase ist höher als beispielsweise bei Classification-based Verfahren und weiterhin hängt die Anomalieerkennungsrate davon ab, wie gut die benutzte Methode zur Distanzbestimmung zwischen normalen und anormalen Daten differenzieren kann.

3.4 Statistische Verfahren

Auch statistische Methoden können zur Erkennung von Anomalien eingesetzt werden. Als Grundlage dazu wird ein statistisches Model benutzt [vgl. BBK14, Seite 9-11]. Im ersten Schritt muss das Model dann mit normalen Daten befüllt werden. Mit Hilfe dieses Models können dann Anomalien erkannt werden, indem geschaut wird, wie wahrscheinlich es ist, dass eine neue Dateninstanz von diesem Model generiert wurde. Bei statistischen Anomalieerkennungsverfahren wird zwischen **parametrisierten** und **nicht-parametrisierten** Varianten unterschieden. Bei der parametrisierten Variante ist die Verteilungsfunktion der Daten schon bekannt und die Parameter werden durch die gegebenen normalen Daten eingestellt. Der Anomalie-Score wird durch das Inverse einer Wahrscheinlichkeitsdichtefunktion $f(x, \Theta)$ mit Parameter Θ und neuer Dateninstanz x bestimmt. Weiterhin unterscheiden sich die parametrisierten Verfahren, darin welche Verteilungsfunktion benutzt wird (z.B. Gaußsche Verteilung oder Regressionsmodell) und der Art und Weise wie Anomalien erkannt werden. Zum Beispiel können in einer einfachen Variante alle Dateninstanzen die mehr als die dreifache Standardabweichung von Durchschnitt der Verteilung entfernt sind, als Anomalien eingeordnet werden oder es wird die Box Plot Rule [LJK00] (siehe Abbildung 3) verwendet. Bei der Box Plot Rule werden durch einen minimalen und einen maximalen Wert Grenzen definiert und der restliche Bereiche wird in unteren, mittleren und oberen Quarter unterteilt. Bei den nicht-parametrisierten Methoden werden statistische Modele ohne Parameter benutzt, wie zum Beispiele Methoden, die auf Histogrammen basieren. Dabei wird zuerst durch die Daten ein Histogramm aufgebaut, das abhängig von den Werten eines Features verschiedene Bereiche mit unterschiedlichen Höhen (stellen Häufigkeiten dar) enthält. Für neue Testinstanzen wird dann überprüft in welchen Bereich des Histogramms der Datensatz fällt. Die Höhe des Bereiches bestimmt dann den Anomalie-Score der Dateninstanz.

Ein Hauptvorteil von statistischen Verfahren ist das ein Anomalie-Score häufig auch eine Wahrscheinlichkeitsangabe als extra Information bereitstellt [vgl. CBK09, Seite 39-40], die angibt wie sicher es sich um eine Anomalie handelt. Weiterhin bieten statistische Verfahren eine statistisch richtige Lösung bei der Erkennung von



Quelle: https://towardsdatascience.com/understanding-boxplots-5e2df7bcbd51

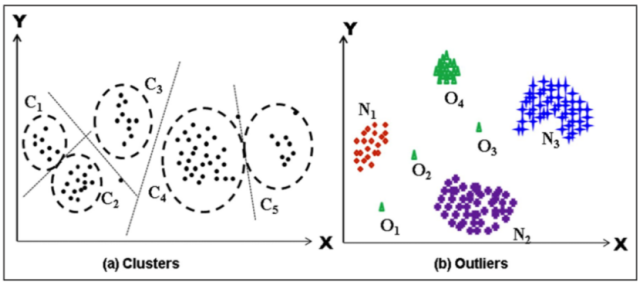
Abb. 3 Abbildung 3 stellt dar wie mit Hilfe eines Boxplots Anomalien identifiziert werden können. Er ist durch die Grenzen Minimum, Q1, Median, Q2 und Maximum in 4 ”Quarter” unterteilt.

Anomalien an, wenn die zugrunde liegende Verteilung stimmt. Es ist auch möglich statistische Verfahren in einem unüberwachten Modus auszuführen. Nachteile sind hingegen, dass statistische Verfahren davon ausgehen, dass die Verteilung der realen Daten auf einer bestimmten Verteilung basiert und diese Annahme nicht immer stimmen muss. Ein weiterer Nachteil ist das Techniken die Histogramme benutzten Schwierigkeiten bei der Erkennung von Anomalien in Daten mit multivariaten Attributen haben.

3.5 Clustering-Verfahren

Ein weitere Möglichkeit zur Anomalieerkennung bieten Clustering-Verfahren. Das Ziel von Clustering ist es eine Menge von ähnlichen Objekten in gemeinsame Cluster einzuordnen [vgl. CBK09, Seite 30]. Abbildung 3 zeigt beispielhaft wie eine Menge von Datenpunkten in fünf verschiedene Cluster eingeordnet werden könnte. Die meisten Clustering-Verfahren operieren im unsupervised Modus. Zur Erkennung von Anomalien beim Clustering können unterschiedliche Annahmen genutzt werden. Eine einfache Annahme ist zum Beispiel, dass nur normale Daten zu den Clustern gehören und Anomalien nicht zu einem Cluster gehören. Algorithmen dieser Kategorie ordnen nicht zwangsläufig jeder Dateninstanz ein Cluster zu und die übriggebliebenen Daten werden als Anomalien eingestuft. Ein weitere Annahme besagt, dass normale Daten dichter am Mittelpunkt des Clusters (cluster centroid) liegen, während anormale Daten weit vom Mittelpunkt entfernt liegen. Ein Algorithmus aus dieser Kategorie wäre zum Beispiel K-means Clustering. Die letzte Annahme geht davon aus, dass normale Dateninstanzen sehr große und dichte Cluster bilden und daher nicht mit kleinen Clustern aus Anomalien zu verwechseln sind.

Die Vorteile von Clustering-Verfahren bei der Anomalieerkennung sind zum einen die Möglichkeit im unsupervised Modus zu operieren [vgl. CBK09, Seite 32] und zum anderen sind Clustering Algorithmen bei der Einordnung von neuen Dateninstanzen sehr schnell, da sie die neue Dateninstanz nur mit einer kleinen Menge von



Quelle: https://ieeexplore.ieee.org/document/6524462 [vgl. BBK14, Seite 14]

Abb. 4 Auf der linke Seite der Abbildung werden insgesamt 5 verschiedene Cluster in einem zweidimensionalen Raum dargestellt. Der rechte Teil verdeutlicht, wo Ausreißer zwischen den Clustern zu finden sind.

Clustern vergleichen müssen. Ein weiterer Vorteil besteht darin, wenn die Anzahl k an Clustern vorab bekannt ist und so wird die Bildung der Cluster und die Zuordnung sehr einfach. Ein Nachteil von Clustering-Verfahren ist, dass einige Clustering Algorithmen Probleme bekommen, wenn Anomalien selber wieder ein Cluster bilden. Ein weiterer Nachteil stellt die hohe Rechenkomplexität $\mathcal{O}(n^2)$ einiger Clustering Algorithmen dar und die Effektivität der Clustering-Verfahren hängt stark davon ab, ob die Cluster die Struktur der normalen Daten abbilden können.

4 Aufbau der Simulationsumgebung

4.1 Verwendete Algorithmen

4.2 Metriken

4.3 Angriffsszenarien

https://ieeexplore.ieee.org/abstract/document/5940552 1) Attack Scenario I - Increased Frequency: 2) Attack Scenario II - Message Flooding: Attack Scenario III - Plausibility of Interrelated Events:

4.4 Daten

5 Durchführung und Ergebnisse des Vergleiches

6 Evaluierung der Ergebnisse

7 Zusammenfassung

Literatur

BBK14. M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita. Network anomaly detection: Methods, systems and tools. *IEEE Communications Surveys Tutorials*, 16(1):303–336, First 2014.

CBK09. Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):15:1–15:58, July 2009.

CMK+11. Stephen Checkoway, Damon Mccoy, Brian Kantor, Danny Anderson, Hovav Shacham, Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, and Tadayoshi Kohno. Comprehensive Experimental Analyses of Automotive Attack Surfaces. *USENIX Security*, 2011.

FYM05. Ryohei Fujimaki, Takehisa Yairi, and Kazuo Machida. An approach to spacecraft anomaly detection problem using kernel feature space. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, KDD '05, pages 401–410, New York, NY, USA, 2005. ACM.

Gmb18. Wirecard Technologies GmbH. Data & analytics – anomalieerkennung in zeitreihen, 2018.

GTDVMFV09. P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez. Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers Security*, 28(1):18 – 28, 2009.

HMVVDK13. Peter Hank, Steffen Müller, Ovidiu Vermesan, and Jeroen Van Den Keybus. Automotive ethernet: In-vehicle networking and smart mobility. In *Proceedings of the Conference on Design, Automation and Test in Europe*, DATE '13, pages 1735–1739, San Jose, CA, USA, 2013. EDA Consortium.

KKSZ09. Hans-Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek. Loop: Local outlier probabilities. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 1649–1652, New York, NY, USA, 2009. ACM.

KNT00. Edwin M. Knorr, Raymond T. Ng, and Vladimir Tucakov. Distance-based outliers: Algorithms and applications. *The VLDB Journal*, 8(3-4):237–253, February 2000.

LJK00. Jorma Laurikkala, Martti Juhola, and Erna Kentala. Informal identification of outliers in medical data. *Fifth International Workshop on Intelligent Data Analysis in Medicine and Pharmacology*, 07 2000.

MA11. M. Müter and N. Asaj. Entropy-based anomaly detection for in-vehicle networks. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 1110–1115, June 2011.

Pat. Andrew Patterson. Outlier detection methods for detecting cheaters in mobile gaming.

Sec. Projekt SecVI. Security for vehicular information - research project for secure automotive network architectures.

SNA+13. I. Studnia, V. Nicomette, E. Alata, Y. Deswarte, M. Kaâniche, and Y. Laarouchi. Survey on security threats and protection mechanisms in embedded automotive networks. In *2013 43rd Annual IEEE/IFIP Conference on Dependable Systems and Networks Workshop (DSN-W)*, pages 1–12, June 2013.