



Machine Learning im Kontext von Cyber Security

Masterarbeit
zur Erlangung des Grades eines Master of Science (M.Sc.) im Studiengang
Informationssysteme

vorgelegt von
Kathi Rodi

Matrikelnummer: 3129378

4. Oktober 2019

Erstgutachter: Prof. Dr. Reinhold von Schwerin

Zweitgutachter: Prof. Dr. Markus Schäffter

Betreuer: Hans-Martin Münch

I. Eigenständigkeitserklärung

Diese Abschlussarbeit wurde von mir selbständig verfasst. Es wurden nur die angegebenen Quellen und Hilfsmittel verwendet. Alle wörtlichen und sinngemäßen Zitate sind in dieser Arbeit als solche kenntlich gemacht.

Kathi Rodi, 4. Oktober 2019

Abstract

Machine Learning Ansätze sind im Kontext von Cyber Security essenziell, da es durch immer anspruchsvoller werdende Sicherheitsbedrohungen nicht mehr möglich ist deren Indikatoren manuell zu ermitteln und zu klassifizieren. Diese Aufgabe von Menschen bearbeiten zu lassen wäre deutlich zu kostenintensiv und zu ineffizient.

Anhand einer Literaturrecherche nach Webster&Watson soll herausgefunden werden welchen Mehrwert Machine Learning in Bezug auf Informationssicherheit bieten kann. Dazu werden bestehende Ansätze aus der Industrie sowohl als auch aus der Wissenschaft klassifiziert, wobei die jeweils verwendeten Algorithmen, Features, Evaluationskriterien sowie die durchgeführte Evaluation der jeweiligen Ergebnisse untersucht werden.

Des Weiteren soll eine Klassifikation bestehender, bzw. eine Formulierung neuer Indicator of Compromise (IoC) zur Unterstützung automatisierter Analysen durchgeführt werden.

Zusätzlich wird recherchiert welche Datensätze in Bezug auf Cyber Security bestehen und welche Qualität diese aufweisen, um einer späteren Machine Learning Anwendung zu dienen.

Um einen übergreifenden Datensatz zusammenführen zu können, welcher unter eine Public License gestellt werden könnte, werden öffentlich zur Verfügung stehende Datensätze analysiert.

Ergänzend werden Diskrepanzen zwischen Ansätzen aus der Industrie und der Wissenschaft überprüft, um etwaige Aussagen aus der Industrie herauszufiltern, welche noch nicht wissenschaftlich belegt wurden. Die Ergebnisse aus diesem Teil der Arbeit könnten zu einem späteren Zeitpunkt dazu dienen, einen Prototypen zu implementieren, welcher einer dieser nicht belegten Aussagen aus der Industrie überprüft. Alternativ könnten gefundene und qualifizierte Datensätze, welche noch keiner Untersuchung unterzogen wurden, einem Prototyp als Basis dienen.

Ziel der Arbeit ist es eine umfassende Übersicht über bestehende Machine Learning Ansätze, welche dabei helfen Indikatoren von Cyberangriffen zu klassifizieren, zu gewinnen und einer dieser gefunden Ansätze beziehungsweise einen gefundenen qualifizierten Datensatz wissenschaftlich zu validieren.

Inhaltsverzeichnis

I.	Eigenständigkeitserklärung	ii
	Inhaltsverzeichnis	iv
1.	Einleitung	2
1.	Motivation	2
2.	Ziel der Arbeit	3
3.	Aufbau der Arbeit	3
2.	Forschungsmethode	4
3.	Theoretische Grundlagen	5
1.	IOCs	5
2.	CRISP-DM	5
4.	Bestehende Analyseverfahren	6
1.	Ansätze aus der Wissenschaft	6
1.1.	Erkennung von Malware - Hybride Analyse (2015)	7
1.2.	Erkennung von Malware - Statische Analyse (2016)	7
1.3.	Erkennung bössartiger XML-basierter Office Dokumente(2016)	8
1.4.	Erkennung bössartiger HTTP-Anfragen (2016)	9
1.5.	Erkennung von Spam Profilen (2016)	9
1.6.	Klassifizierung von Netzwerkattacken (2017)	9
1.7.	Erkennung von Ransomware - Dynamische Analyse (2017) .	9
1.8.	Erkennung von Malware - Imageanalyse (2017)	9

1.9.	Erkennung böswilliger MS Office Macros (2017)	9
1.10.	Klassifizierung von Distributed Denial of Service (DDoS) At- tacken (2017)	10
1.11.	Schwachstellen Scanner für Web Applikationen (2017)	10
1.12.	Erkennung von Malware anhand von PE-Header (2017)	10
1.13.	Erkennung von Malware anhand von PE-Header mit erwei- tertem Feature-Set (2017)	10
1.14.	Erkennung von Exfiltration und Command & Control (C&C)Tunnels (2017)	10
1.15.	Erkennung bössartiger PowerShell-Befehle (2018)	10
1.16.	Klassifizierung von Netzwerkverkehr in 5 Klassen (2018) . .	10
1.17.	Anomalieerkennung anhand von Systemprotokollen (2018) .	10
1.18.	Erkennung von bössartigem Netzwerkverkehr (2018)	11
1.19.	Erkennung von Botnetzen (2018)	11
1.20.	Klassifizierung von Microsoft Malware (2018)	11
1.21.	Klassifizierung von Malware anhand von Datenpaketen (2018)	11
1.22.	Erkennung von Portscans (2018)	11
1.23.	Klassifizierung von Netzwerkverkehr in 3 Klassen (2018) . .	11
1.24.	Erkennung bössartiger SQL-Abfragen (2018)	11
1.25.	Erkennung von Low-rate DDoS (LDDoS) Attacken (2018) . .	11
1.26.	Klassifizierung von Wi-Fi Netzwerkdaten (2018)	12
1.27.	Klassifizierung von Malware anhand einer systematischen Profilerstellung (2019)	12
1.28.	Klassifizierung von Malware - Imageanalyse (2019)	12
1.29.	Erkennung von Fast-Flux (FF) Netzwerken (2019)	12
1.30.	Erkennung von drive-by Download-Attacken bei Twitter (2019)	13
1.31.	Erkennung von Domain Generation Algorithm (DGA) Do- mains (2019)	13
1.32.	Erkennung von Phishing Websites (2019)	13

1.33. Erkennung von Insider Bedrohungen (2019)	13
1.34. Erkennung von bösartigen PDFs (2019)	13
2. Ansätze aus der Industrie	14
3. Diskussion	14
5. Klassifizierung von IOCs	15
6. Datensätze	16
7. Fazit	17
8. Zukünftige Forschung	18
A. Literatur	I
Abbildungsverzeichnis	V
Tabellenverzeichnis	VI
Abkürzungsverzeichnis	VI

1. Einleitung

Bereits im 19. Jahrhundert träumte der Polymath Charles Babbage vom mechanisierten Rechnen. Dieser Wunsch basierte hauptsächlich auf dem Zorn über die Unzulänglichkeit der damaligen analogen mathematischen Anwendungen. Babbage entwickelte ein Konzept für analytische Maschinen, also einen programmierbaren Allzweckrechner. Seine Kollegin, die britische Mathematikerin, Ada Lovelace lieferte die entsprechenden Ideen zur Programmierung seiner Maschine. Allerdings konnte das Konzept der *Analytical Engine* niemals umgesetzt werden und besteht seither, rein als Entwurf. Dennoch macht diese Forschung die beiden bis heute zu Pionieren des modernen Computers und dessen Programmierung.

Babbages Wunschtraum von damals ist nicht nur längst Wirklichkeit geworden, er hat sich in rasendem Tempo weiterentwickelt. Heute können Computer nicht nur fehlerfrei Logarithmen berechnen, sie sind selbst in der Lage einen Großteil unseres Lebens zu digitalisieren. Bankgeschäfte, Einkäufe, die Steuererklärung und bald auch Arztbesuche sind nur ein kleiner Teil dessen, was wir online erledigen. Dabei produzieren wir eine enorme Masse an persönlichen Daten, welche in falschen Händen, zu einem physischen Schaden für uns führen können. Gerade deshalb gilt es diesen Teil unseres Lebens zu schützen. Wie wir unsere physischen Habseligkeiten schützen in dem wir beispielsweise Schlösser verwenden, gilt es ebenso unsere digitalen Artefakte zu schützen, um finanziellen, reputativen sowie physischen Schaden zu verhindern. Studien zeigen allerdings, dass wir der nötigen Sicherheit weit hinterherhinken.

1. Motivation

Cybercrime umfasst die Straftaten, die sich gegen Datennetze, informationstechnische Systeme oder deren Daten richten [...] oder die mittels Informationstechnik begangen werden. (Bundeskriminalamt 2018)

Das BKA verzeichnete allein im Jahr 2017 knapp 86.000 Fälle von Cybercrime. Davon waren über 1.400 Phishing Angriffe im Onlinebanking bei denen ein durchschnittlicher Schaden von 4000 € pro Fall entstand (Bundeskriminalamt 2018). Laut einer Studie des Bundesverband Informationswirtschaft, Telekommunikation

und neue Medien e.V. (BITKOM) ist bereits, jeder zweite Deutsche Opfer eines Cyberangriffs geworden, lediglich 18% hätten diesbezüglich angegeben, Anzeige bei der Polizei erstattet zu haben (e.V. 2017). Dies lässt vermuten, dass die Dunkelziffer der tatsächlichen Cybercrime-Straftaten weit über den 86.000 gemeldeten Fällen liegt.

2. Ziel der Arbeit

3. Aufbau der Arbeit

2. Forschungsmethode

3. Theoretische Grundlagen

1. IOCs

2. CRISP-DM

4. Bestehende Analyseverfahren

Das nun folgende Kapitel beschäftigt sich mit Machine Learning Verfahren, welche für die Erkennung von Cyber Security Angriffen verwendet werden. Diese Verfahren wurden anhand einer umfangreichen Literaturrecherche ermittelt. Jedes Verfahren wird auf dessen verwendete Algorithmen sowie der ausgewählten Features untersucht. Des Weiteren wird analysiert welche Evaluationskriterien verwendet wurden und zu welchem Ergebnis der jeweilige Ansatz führte.

Zunächst werden die Verfahren erläutert, welche bereits wissenschaftlich belegt wurden. Im Anschluss werden Verfahren aus der Industrie untersucht.

1. Ansätze aus der Wissenschaft

Im folgenden Abschnitt werden die durch die Literaturrecherche ermittelten Analyseverfahren aus der Wissenschaft dokumentiert. Um die Entwicklung der Ansätze im Laufe der Zeit beobachten zu können, werden die Verfahren aufsteigend nach Jahreszahl erläutert. Für die Recherche wurden alle Verfahren in einem Zeitraum von 2015 bis heute berücksichtigt. Dieser Zeitraum wurde gewählt, da sich die Zahl der Cyberangriffe, sowie die zur Verfügung stehende Schadsoftware bereits innerhalb eines Jahres deutlich vermehrt beziehungsweise verändert. Somit soll verhindert werden veraltete Angriffsvektoren zu analysieren, welche bereits von moderneren überholt wurden. Zusätzlich gibt es bereits vergleichbare Arbeiten aus dem Jahr 2016 (s. Buczak und Guven 2016), in welchem Machine Learning Ansätze vor dieser Zeit analysiert werden.

In der folgenden Auflistung wird die Bezeichnung „Erkennung“ für binäre Klassifikation verwendet. Beispielsweise, wenn sich ein Ansatz darauf beschränkt Daten entweder in die Rubrik A „böartig“ oder in die Rubrik B „gutartig“ zu klassifizieren. Erfolgt in einem Analyseverfahren eine Einteilung in mehrere Klassen (mehr als zwei), wird nachfolgend der Begriff „Klassifizierung“ verwendet.

1.1. Erkennung von Malware - Hybride Analyse (2015)

Im Jahr 2015 haben Shijo und Salim einen, auf zwei Analysen basierenden, Ansatz gewählt, um Malware zu erkennen. Zum einen verwendeten sie ein statisches Analyseverfahren bei dem sie *Printable Strings*, also nicht kodierte Zeichenfolgen wie z. B. *FindFirstFile* aus Binärdateien extrahierten. Zum anderen konfigurierten sie eine Cuckoo Sandbox, in der sie Schadsoftware ausführten und deren API Aufrufe in einer Logdatei speicherten.

Sie untersuchten die Ähnlichkeit in API-Aufrufsequenzen anhand von n-Gramm-basierter Ähnlichkeitsmessung. Als Features dienten Tri- und Tetragramme ab einer gewissen Häufigkeit, sowie PrintableStrings ab einer Häufigkeit von zwei. Für die Klassifizierung wurden die Algorithmen Random Forest (RF) und Support Vector Machine (SVM) verwendet. Es wurden jeweils beide Ansätze separat, sowie in Kombination getestet. Analysen mit SVM erzielten eine Genauigkeit von 95.88 % für die statische Analyse und 97.16 % für die dynamische Analyse und waren somit erfolgreicher, als Untersuchungen mit Random Forest. Die besten Ergebnisse erzielte der hybride Ansatz mit SVM mit einer Genauigkeit von 98.71 % und der geringsten False Positive Rate (FPR) von 0.026.

Die Forschung von Shijo und Salim zeigt also, dass mit den von ihnen gewählte Features, mit einem hybriden Ansatz, deutlich genauere Aussagen, als mit rein statischen oder rein dynamischen Analysen, getroffen werden können.

1.2. Erkennung von Malware - Statische Analyse (2016)

More und Gaikwad untersuchten EXE-Dateien auf Schadsoftware. Dazu konvertierten sie die Dateien zunächst in Operation Code (Opcode), also in den Teil der Maschinensprachanweisung der die auszuführenden Operationen angibt, z.B. 55 8B EC 83 EC 5C 83 7D 0C 0F 74 2B 83 7D 0C 46. Das ausgewählte Feature Datenset wurde anschließend nochmals zu einer Attribute-Relation File Format (ARFF) Datei konvertiert, um die Datei nachfolgend mit der Machine Learning Software Weka bearbeiten zu können. In Weka wurden die Algorithmen JRip, C4.5 und IBk verwendet. Wobei es sich bei JRip und C4.5 um Decision Tree (DT) und bei IBk um k-nearest-neighbor (k-NN) Implementierungen handelt. Um die Erkennungsgenauigkeit zu erhöhen, wurden nicht die einzelnen Algorithmen, sondern ein Klassifikatorenensemble angewandt, um Methoden wie Mehrheitsvoting, Veto-Voting und vertrauensbasiertes Veto-Voting verwenden zu können. Ersteres folgt demokratischen Regeln, das heißt, die Klasse mit den meisten Stimmen ist das Ergebnis. Veto-Voting hingegen basiert auf Annahmen über die Wahl der anderen Algorithmen. Vertrauensbasiertes Veto-Voting ergänzt voriges Voting um eine Vertrauensberechnung, wodurch jedem Algorithmus ein bestimmtes Vertrauensniveau

zugeteilt wird.

More und Gaikwad konnten zeigen, dass durch die Verwendung von Veto-Voting eine Genauigkeit von 80.7 % erzielt werden kann. Im Vergleich dazu, lag das beste Ergebnis, welches durch singulären Algorithmen Einsatz von IBk erzielt wurde, bei einer Genauigkeit von nur 73.5 %.

Dieses Ergebnis stützt die These von Shijo und Salim aus dem Jahr zuvor, welche zeigten, dass ein nicht-hybrider Ansatz weniger genau ist, als einer, der die statische und die dynamische miteinander verknüpft.

1.3. Erkennung bössartiger XML-basierter Office Dokumente(2016)

Durch das neue Dateiformat, welches Microsoft 2007 auf den Markt gebracht hat, sollten Sicherheitslücken geschlossen werden. Das Binärformat wurde durch ein XML basiertes Dateiformat ersetzt. Dadurch werden neue digitale Funktionen unterstützt, sowie vertrauensbasierte Bereiche geschaffen, welche das Format weniger riskoreich gestalten sollen. Nichtsdestotrotz können Attacken gegen XML-basierte Office Dokumente gestartet werden. Zu den möglichen Angriffsvektoren zählen beispielsweise macrobasierte Attacken. Durch den Missbrauch von Visual Basic for Applications (VBA) kann die zugehörige shell gestartet werden, um willkürliche Kommandos zu senden. Ausserdem können externe Bibliotheken sowie Programme aufgerufen werden, welche Schaden verursachen können. Eine weitere Bedrohung durch den Gebrauch von Macros bildet die Fähigkeit dieser, bössartige Dateien aus dem Internet herunterzuladen.

Cohen u. a. (2016) haben in ihrer Arbeit diese Art von Attackvektoren untersucht, um eine Infizierung durch bössartige Macros frühzeitig zu erkennen. Dazu haben sie zunächst Office Dokumente in eine Liste von Pfaden konvertiert. Diese dienen der Analyse als Features. Dadurch wurde jedoch eine so hohe Anzahl an Features generiert, dass die Untersuchung mit verschiedenen Datensets durchgeführt wurde, welche Top Features von 10 bis 2000 beinhalteten. Um die Feature-Repräsentation, also das Vorhandensein bzw. die Wichtigkeit von Features zu bestimmen, wurden zwei Verfahren angewandt. Zum einen ein binäres Verfahren, welches lediglich die Ab-, respektive Anwesenheit eines Features misst und zum anderen wurde das statistische Verfahren Term Frequency - Inverse Document Frequency (TF-IDF) verwendet, um die Wichtigkeit eines Terms in Bezug auf ein Dokument zu bestimmen. Anschließend wurden die Daten mit folgenden Algorithmen untersucht: J48, RF, Logistic Regression (LR), Naïve Bayes (NB), Bayesian Network (BN), LogitBoost (LB), Sequential Minimal Optimization (SMO), Bagging und AdaBoost (AB).

Wie die Ergebnisse zeigen, erzielt das Datenset mit den Top 200 Fetaures, welches mit Random Forest analysiert wurde die besten Werte mit einem F-measure von 0.66. Wie sich demonstrieren ließ, ist es möglich böartige Office Dokumente durch eine Analyse deren Pfade zu erkennen. Die Untersuchung beschränkt sich jedoch auf direkte Gefahren innerhalb von Dokumenten. Indirekte Gefahren, wie etwa die durch weiterführende Links, wurden in dieser Forschungsarbeit nicht berücksichtigt.

1.4. Erkennung böartiger HTTP-Anfragen (2016)

Pham, Hoang und Vu (2016)... anhand des Datensets: Carmen Torrano Giménez, Alejandro Pérez Villegas (2010)...

1.5. Erkennung von Spam Profilen (2016)

Singh, Bansal und Sofat (2016)...

1.6. Klassifizierung von Netzwerkattacken (2017)

Yin u. a. (2017)....

1.7. Erkennung von Ransomware - Dynamische Analyse (2017)

Maniath u. a. (2017)...

1.8. Erkennung von Malware - Imageanalyse (2017)

Roma2017...

1.9. Erkennung böswilliger MS Office Macros (2017)

Bearden und Lo (2017)....

1.10. Klassifizierung von DDoS Attacken (2017)

Z. He, Zhang und Lee (2017)...

1.11. Schwachstellen Scanner für Web Applikationen (2017)

Robin Tommy, Gullapudi Sundeep (2017)...

1.12. Erkennung von Malware anhand von PE-Header (2017)

Raff, Sylvester und Nicholas (2017)...

1.13. Erkennung von Malware anhand von PE-Header mit erweitertem Feature-Set (2017)

Kumar, Kuppusamy und Aghila (2017)...

1.14. Erkennung von Exfiltration und C&CTunnels (2017)

Das u. a. (2017)...

1.15. Erkennung bössartiger PowerShell-Befehle (2018)

Hendler, Kels und Rubin (2018)...

1.16. Klassifizierung von Netzwerkverkehr in 5 Klassen (2018)

Ding und Zhai (2018)...

1.17. Anomalieerkennung anhand von Systemprotokollen (2018)

Brown u. a. (2018)...

1.18. Erkennung von böartigem Netzwerkverkehr (2018)

Aldwairi, Perera und Novotny (2018)...

1.19. Erkennung von Botnetzen (2018)

Mathur, Raheja und Ahlawat (2018)....

1.20. Klassifizierung von Microsoft Malware (2018)

Sabar, Yi und Song (2018)....

1.21. Klassifizierung von Malware anhand von Datenpaketen (2018)

Yeo u. a. (2018)...

1.22. Erkennung von Portscans (2018)

Aksu und Ali Aydin (2018)

1.23. Klassifizierung von Netzwerkverkehr in 3 Klassen (2018)

Teoh u. a. (2018)...

1.24. Erkennung böartiger SQL-Abfragen (2018)

Jayaprakash und Kandasamy (2018)...

1.25. Erkennung von LDDoS Attacken (2018)

Siracusano, Shiaeles und Ghita (2018)...

1.26. Klassifizierung von Wi-Fi Netzwerkdaten (2018)

Qin u. a. (2018)....

1.27. Klassifizierung von Malware anhand einer systematischen Profilerstellung (2019)

P. He u. a. (2017) Malware Verschleierung durch packing, Metamorphism und Polymorphism.

Han u. a. (2019)...

..... Neueste Forschungen zeigen, dass dynamische und statische Analyseverfahren zu ungenau sind, um neue Schadsoftware in Echtzeit zu erkennen (Vinayakumar u. a. 2019). Dazu bedarf es Deep Learning Verfahren, wie der folgende Ansatz verdeutlicht.

1.28. Klassifizierung von Malware - Imageanalyse (2019)

Um Polymorphismus, Metamorphismus und Packing zu erkennen ist ein umfangreiches Feature Engineering, sowie beträchtliche Kenntnisse auf Domain-Ebene nötig (Rhode, Burnap und Jones 2018). Zudem können Angreifer der automatischen Malware Erkennung entgehen sobald sie die verwendeten Features kennen (Anderson u. a. 2017). Diesen Problemen wollen Vinayakumar u. a. (2019) mit ihrem Deep Learning Ansatz begegnen.

Dazu verglichen sie klassische Algorithmen für maschinelles Lernen mit Deep Learning Architekturen. Die Vergleiche basieren auf statischen und dynamischen Analysen, sowie auf Bildverarbeitungstechnologien.

...

Wie sich zeigte ist die Imageanalyse schneller als die statische und die dynamische Analyse, da diese auf Rohdaten basiert und komplett auf Zerlegung oder Ausführung von Code verzichtet.

1.29. Erkennung von FF Netzwerken (2019)

Chen u. a. (2019)...

1.30. Erkennung von drive-by Download-Attacken bei Twitter (2019)

Javed, Burnap und Rana (2019)... Verschleierte Links können dazu führen dass der Angreifer Fernzugriff zum System des Opfers bekommt von welchem er Daten klauen kann oder dessen Computer in ein Botnetz integrieren kann (Provost u. a. 2007).

1.31. Erkennung von DGA Domains (2019)

Li u. a. (2019)...

1.32. Erkennung von Phishing Websites (2019)

Phishing zielt im Vergleich zu anderen Attacken nicht darauf ab Schwachstellen im System auszunutzen, sondern durch Sicherheitslücken beim Menschen an dessen sensitive Informationen wie Benutzernamen und Passwörter zu gelangen. In der Forschung gibt es momentan vier Verfahren, um Phishing Websites zu erkennen: Blacklists, Heuristiken, Inhaltsanalysen und Maschinelles Lernen. Blacklists gleichen URLs mit bekannten Phishing Websites ab, Heuristiken verwenden Signaturdatenbanken bekannter Angriffe um sie mit der Signatur eines heuristischen Musters abzugleichen. Inhaltsanalysen versuchen Phishing Websites mit Hilfe bekannter Algorithmen wie TF-IDF zu identifizieren.

Der im Folgende beschriebene Ansatz von Alswailem u. a. (2019) verwendet Machine Learning Verfahren, um Phishing Websites zu erkennen.

1.33. Erkennung von Insider Bedrohungen (2019)

Le und Nur Zincir-Heywood (2019)...

1.34. Erkennung von böartigen PDFs (2019)

Jeong, Woo und Kang (2019)...

2. Ansätze aus der Industrie

Microsoft (2019)

3. Diskussion

5. Klassifizierung von IOCs

6. Datensätze

7. Fazit

8. Zukünftige Forschung

A. Literatur

Bücher und Journals

- Aksu, D. und M. Ali Aydin (2018). “Detecting Port Scan Attempts with Comparative Analysis of Deep Learning and Support Vector Machine Algorithms”. In: *2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT)*. IEEE, S. 77–80 (siehe S. 11).
- Aldwairi, T., D. Perera und M. A. Novotny (2018). “An evaluation of the performance of Restricted Boltzmann Machines as a model for anomaly network intrusion detection”. In: *Computer Networks* 144, S. 111–119 (siehe S. 11).
- Alswailem, A. u. a. (Mai 2019). “Detecting Phishing Websites Using Machine Learning”. In: *2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*. IEEE, S. 1–6 (siehe S. 13).
- Anderson, H. S. u. a. (2017). “Evading machine learning malware detection”. In: *Black Hat* (siehe S. 12).
- Bearden, R. und D. C.-T. Lo (2017). “Automated microsoft office macro malware detection using machine learning”. In: *2017 IEEE International Conference on Big Data (Big Data)*. Bd. 2018-Janua. IEEE, S. 4448–4452 (siehe S. 9).
- Brown, A. u. a. (2018). “Recurrent Neural Network Attention Mechanisms for Interpretable System Log Anomaly Detection”. In: *Proceedings of the First Workshop on Machine Learning for Computing Systems*, S. 1 (siehe S. 10).
- Buczak, A. L. und E. Guven (2016). “A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection”. In: *IEEE Communications Surveys & Tutorials* 18.2, S. 1153–1176 (siehe S. 6).
- Bundeskriminalamt (2018). “Cybercrime, Bundeslagebild 2017”. In: (Siehe S. 2).
- Chen, X. u. a. (2019). “A Deep Learning Based Fast-Flux and CDN Domain Names Recognition Method”. In: *Proceedings of the 2019 2nd International Conference on Information Science and Systems - ICISS 2019*. Bd. Part F1483. New York, New York, USA: ACM Press, S. 54–59 (siehe S. 12).
- Cohen, A. u. a. (2016). “SFEM: Structural feature extraction methodology for the detection of malicious office documents using machine learning methods”. In: *Expert Systems with Applications* 63, S. 324–343 (siehe S. 8).

- Das, A. u. a. (Dez. 2017). “Detection of Exfiltration and Tunneling over DNS”. In: *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. Bd. 2018-Janua. IEEE, S. 737–742 (siehe S. 10).
- Ding, Y. und Y. Zhai (2018). “Intrusion Detection System for NSL-KDD Dataset Using Convolutional Neural Networks”. In: *Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence - CSAI '18*. New York, New York, USA: ACM Press, S. 81–85 (siehe S. 10).
- e.V., B. (2017). *Cybercrime: Jeder zweite Internetnutzer wurde Opfer*. URL: <https://www.bitkom.org/Presse/Presseinformation/Cybercrime-Jeder-zweite-Internetnutzer-wurde-Opfer.html> (besucht am 01.10.2019) (siehe S. 3).
- Han, W. u. a. (Jan. 2019). “MalInsight: A systematic profiling based malware detection framework”. In: *Journal of Network and Computer Applications* 125.June 2018, S. 236–250 (siehe S. 12).
- He, P. u. a. (2017). “Model approach to grammatical evolution: deep-structured analyzing of model and representation”. In: *Soft Computing* 21.18, S. 5413–5423 (siehe S. 12).
- He, Z., T. Zhang und R. B. Lee (2017). “Machine Learning Based DDoS Attack Detection from Source Side in Cloud”. In: *2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud)*. IEEE, S. 114–120 (siehe S. 10).
- Hendler, D., S. Kels und A. Rubin (2018). “Detecting Malicious PowerShell Commands using Deep Neural Networks”. In: *Proceedings of the 2018 on Asia Conference on Computer and Communications Security - ASIACCS '18*. New York, New York, USA: ACM Press, S. 187–197 (siehe S. 10).
- Javed, A., P. Burnap und O. Rana (Mai 2019). “Prediction of drive-by download attacks on Twitter”. In: *Information Processing & Management* 56.3, S. 1133–1145 (siehe S. 13).
- Jayaprakash, S. und K. Kandasamy (Apr. 2018). “Database Intrusion Detection System Using Octaplet and Machine Learning”. In: *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*. Iicct. IEEE, S. 1413–1416 (siehe S. 11).
- Jeong, Y.-S., J. Woo und A. R. Kang (Apr. 2019). “Malware Detection on Byte Streams of PDF Files Using Convolutional Neural Networks”. In: *Security and Communication Networks* 2019, S. 1–9 (siehe S. 13).
- Kumar, A., K. Kuppusamy und G. Aghila (2017). “A learning model to detect maliciousness of portable executable using integrated feature set”. In: *Journal of King Saud University - Computer and Information Sciences* 31.2, S. 252–265 (siehe S. 10).

- Le, D. C. und A. Nur Zincir-Heywood (2019). “Machine learning based insider threat modelling and detection”. In: *2019 IFIP/IEEE Symposium on Integrated Network and Service Management, IM 2019*, S. 1–6 (siehe S. 13).
- Li, Y. u. a. (2019). “A Machine Learning Framework for Domain Generation Algorithm-Based Malware Detection”. In: *IEEE Access* 7, S. 32765–32782 (siehe S. 13).
- Maniath, S. u. a. (2017). “Deep learning LSTM based ransomware detection”. In: *2017 Recent Developments in Control, Automation & Power Engineering (RDCAPE)*. Bd. 3. IEEE, S. 442–446 (siehe S. 9).
- Mathur, L., M. Raheja und P. Ahlawat (2018). “Botnet Detection via mining of network traffic flow”. In: *Procedia Computer Science* 132, S. 1668–1677 (siehe S. 11).
- More, S. S. und P. P. Gaikwad (2016). “Trust-based Voting Method for Efficient Malware Detection”. In: *Procedia Computer Science* 79, S. 657–667 (siehe S. 7, 8).
- Pham, T. S., T. H. Hoang und V. C. Vu (2016). “Machine learning techniques for web intrusion detection — A comparison”. In: *2016 Eighth International Conference on Knowledge and Systems Engineering (KSE)*. IEEE, S. 291–297 (siehe S. 9).
- Provos, N. u. a. (2007). “The Ghost in the Browser: Analysis of Web-based Malware.” In: *HotBots* 7, S. 4 (siehe S. 13).
- Qin, Y. u. a. (Sep. 2018). “Attack Detection for Wireless Enterprise Network: a Machine Learning Approach”. In: *2018 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*. IEEE, S. 1–6 (siehe S. 12).
- Raff, E., J. Sylvester und C. Nicholas (2017). “Learning the PE Header, Malware Detection with Minimal Domain Knowledge”. In: *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security - AISec '17*. New York, New York, USA: ACM Press, S. 121–132 (siehe S. 10).
- Rhode, M., P. Burnap und K. Jones (2018). “Early-stage malware prediction using recurrent neural networks”. In: *computers & security* 77, S. 578–594 (siehe S. 12).
- Robin Tommy, Gullapudi Sundeep, H. J. (2017). “Automatic Detection and Correction of Vulnerabilities using Machine Learning”. In: *2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC)*, S. 1062–1065 (siehe S. 10).
- Sabar, N. R., X. Yi und A. Song (2018). “A Bi-objective Hyper-Heuristic Support Vector Machines for Big Data Cyber-Security”. In: *IEEE Access* 6, S. 10421–10431 (siehe S. 11).

- Shijo, P. und A. Salim (2015). “Integrated Static and Dynamic Analysis for Malware Detection”. In: *Procedia Computer Science* 46.Icict 2014, S. 804–811 (siehe S. 7, 8).
- Singh, M., D. Bansal und S. Sofat (2016). “A Novel Technique to Characterize Social Network Users: Comparative Study”. In: *Proceedings of the 6th International Conference on Communication and Network Security*. ACM, S. 75–79 (siehe S. 9).
- Siracusano, M., S. Shiaeles und B. Ghita (2018). “Detection of LDDoS Attacks Based on TCP Connection Parameters”. In: *2018 Global Information Infrastructure and Networking Symposium (GIIS)*. IEEE, S. 1–6 (siehe S. 11).
- Teoh, T. T. u. a. (2018). “Anomaly detection in cyber security attacks on networks using MLP deep learning”. In: *2018 International Conference on Smart Computing and Electronic Enterprise (ICSCEE)*. IEEE, S. 1–5 (siehe S. 11).
- Vinayakumar, R. u. a. (2019). “Robust Intelligent Malware Detection Using Deep Learning”. In: *IEEE Access* 7, S. 46717–46738 (siehe S. 12).
- Yeo, M. u. a. (2018). “Flow-based malware detection using convolutional neural network”. In: *2018 International Conference on Information Networking (ICOIN)*. Bd. 2018-Janua. IEEE, S. 910–913 (siehe S. 11).
- Yin, C. u. a. (2017). “A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks”. In: *IEEE Access* 5, S. 21954–21961 (siehe S. 9).

Internetquellen

- Carmen Torrano Giménez, Alejandro Pérez Villegas, G. Á. M. (2010). *CSIC 2010 HTTP dataset*. URL: <http://www.isi.csic.es/dataset/> (siehe S. 9).
- Microsoft (2019). *MICROSOFT SECURITY INTELLIGENCE REPORT*. Techn. Ber. URL: <https://clouddamcdnprodep.azureedge.net/gdc/gdc6fw6T1/original> (siehe S. 14).

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

IoC Indicator of Compromise.....	iii
DDoS Distributed Denial of Service.....	v
LDDoS Low-rate DDoS.....	v
C&C Command & Control.....	v
FF Fast-Flux.....	v
DGA Domain Generation Algorithm.....	v
TF-IDF Term Frequency - Inverse Document Frequency.....	8
RF Random Forest.....	7
SVM Support Vector Machine.....	7
FPR False Positive Rate.....	7
Opcode Operation Code.....	7
ARFF Attribute-Relation File Format.....	7

DT Decision Tree	7
k-NN k-nearest-neighbor	7
VBA Visual Basic for Applications	8
SMO Sequential Minimal Optimization	8
LR Logistic Regression	8
NB Naïve Bayes	8
BN Bayesian Network	8
LB LogitBoost	8
AB AdaBoost	8
BITKOM Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e.V.	2