Masterarbeit Kolloquium

# Machine Learning im Kontext von Cyber Security

**Studiengang: Informationssysteme**


**Verfasserin:** **Kathrin Rodi**
Matrikel-Nr.: 3129378

**Erstgutachter:** **Prof. Dr. Reinhold von Schwerin**
**Zweitgutachter:** **Prof. Dr. Markus Schäffter**
**Betreuer:** **Hans-Martin Münch**

# Agenda

1. Motivation
2. Ziel
3. Konzept
4. IoCs
5. Analyseverfahren
6. Datensätze
7. Prototyp
8. Fazit und Ausblick

# MOTIVATION

# 1. Motivation

- Täglich neue Malware im 6-stelligen Bereich (AV-Test 2019)
- Höchstand neuer Malware (McAfee 2019)

  Quantität
  Qualität

**Machine Learning**

- Binäre/Multi-Klassen-Klassifikation
- Clustering
- Deep Learning

# ZIEL

# 2. Ziel

- Was sind **IoCs** in Bezug auf Malware ?
- Welche **Machine Learning** Verfahren werden in der **Cyber Security** angewendet ?
- Welche **Datensätze** gibt es ?
- Implementierung eines **Prototyps**

# KONZEPT

# 3 Konzept

| Ziel | Methode | Erwartetes Ergebnis |
|---|---|---|
| IoCs | Literaturrecherche | Definition inklusive Beispiele |
| Analyseverfahren | Literaturrecherche | Übersicht diverser Ansätze |
| Datensätze | Literaturrecherche | Evaluierte Datensätze |
| Prototyp | CRISP-DM | Evaluierte Modelle |

# IoCs

# 4.1 IoCs - Definition

Indicators of Compromise (IoCs)

# 4.1 IoCs - Definition

## Indicators of Compromise (IoCs)

Hinterlassenschaften welche ein Angreifer zurück lässt, die darauf hindeuten, dass ein System kompromittiert wurde.

# 4.2 IoCs in Bezug auf Malware

## Portable Executable Dateien (PE-Dateien)

**Dynamische Analyse**

Verdächtige API Aufrufe

**Statische Analyse**

Verdächtige Imports/Exports/Ressourcen/Kompilierungszeit

# ANALYSEVERFAHREN

# 5.1 Analyseverfahren - Vorgehen

33 Ansätze untersucht auf:

- Gegenstand
- Algorithmen
- Features
- Leistungsmetriken
- Ergebnis

# 5.2 Analyseverfahren - Ergebnis

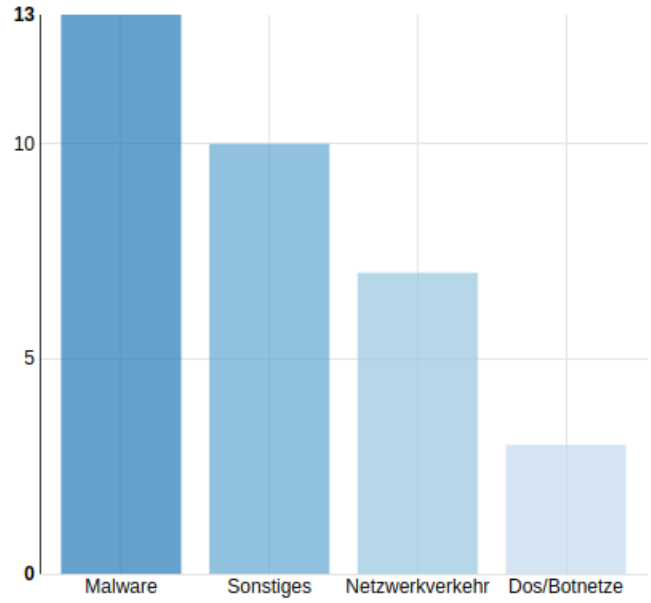


*Abb. 1: Verteilung der Analyseverfahren nach Themen*
*(eigene Darstellung)*

- Trend: Malware
- Features: Use Case spezifisch
- 9 Leistungsmetriken: Accuracy
- 33 Algorithmen
  - 25 traditionelle MLAs: DT, RF
  - 8 Deep Learning: CNN, RNN

# DATENSÄTZE

# 6. Datensätze

## 11 Datensätze verglichen auf:

- Jahr
- Inhalt
- Features
- Umfang
- Labels

# 6. Datensätze

11 Datensätze verglichen auf:

- Jahr
- Inhalt
- Features
- Umfang
- Labels

Ember Datenset

# PROTOTYP

# 7.1 Prototyp - Modeling

- 3 LSTM Modelle
- Keras API in Google Colab
- Mit GPU und Tensorflow Backend
- Trainingszeit max: 500 Epochen
- Leistungsmetriken: Accuracy, loss

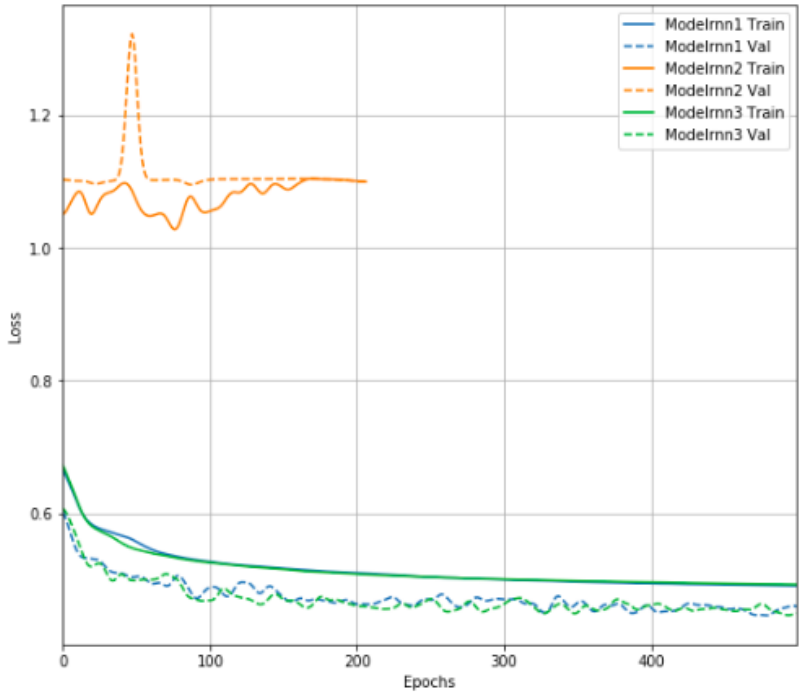


*Abb. 2: Loss nach Epochen*
*(eigene Darstellung)*

# 7.2 Prototyp Evaluierung

Bestes Modell: **modelRnn1**

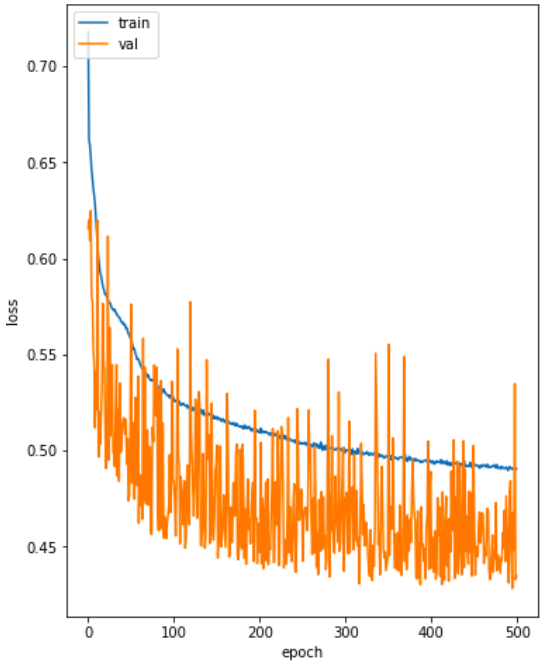


*Abb. 3: loss modelRnn1*
*(eigene Darstellung)*

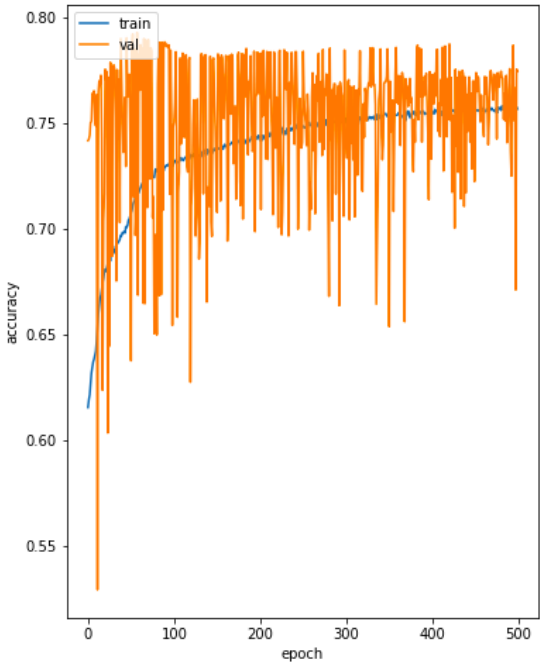


*Abb. 4: Accuracy modelRnn1*
*(eigene Darstellung)*

Accuracy: 78.7%, loss: 0.49

# FAZIT & AUSBLICK

# 8. Fazit und Ausblick

- IoCs dienen als Features
  Weitere Forschung über PE- Dateien hinaus

# 8. Fazit und Ausblick

- IoCs dienen als Features
  Weitere Forschung über PE- Dateien hinaus
- Breite Adaption an Analyseverfahren
  Hilfestellungen für die Industrie bieten

# 8. Fazit und Ausblick

- IoCs dienen als Features
  Weitere Forschung über PE- Dateien hinaus
- Breite Adaption an Analyseverfahren
  Hilfestellungen für die Industrie bieten
- Mangel an Datensets
  Standardisierung

# 8. Fazit und Ausblick

- IoCs dienen als Features
  Weitere Forschung über PE- Dateien hinaus
- Breite Adaption an Analyseverfahren
  Hilfestellungen für die Industrie bieten
- Mangel an Datensets
  Standardisierung
- Erfolgreicher Prototyp
  Basis für weitere Forschung

Vielen Dank für die Aufmerksamkeit

Fragen?

# Quellenverzeichnis

## Internetquellen


- AV-TEST (2019). Malware Statistics & Trends Report.
  url: https://www.avtest.org/en/statistics/malware/
  (besucht am 08. 10. 2019)

- McAfee (2019). McAfee Labs Threats Report.
  url: https://www.mcafee.com/enterprise/en-us/threat-center/mcafee-labs/reports.html
  (besucht am 19. 12. 2019)


## Bücher


- Raschka, S. und V. Mirjalili (2019). Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2. Packt Publishing Ltd

# BACKUP

# Verschleierungsansätze

1. Packing

   Komprimierung exekutierbarer Dateien

2. Metamorphismus

   Auomatische Neukodierung

3. Polymorphismus

   Permanente Veränderung oder Weiterentwicklung von Malware

# IoCs

## Eindeutige IoCs

Malware inklusive von Malware ausgelöste Aktionen

## Nicht eindeutige IoCs

Administrative Tools wie **PsExec**

# PE-Dateien

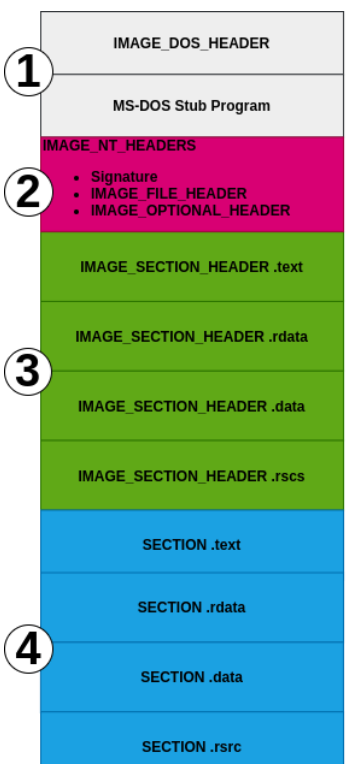


*Abb. 5: Aufbau einer PE-Datei*
*(eigene Darstellung)*

1. Irrelevant für Analyse
2. Grundlegende Informationen der Datei z. B. Kompilierungszeit
3. Programmeinstiegspunkt, Stackgröße, GUI oder Konsole
4. Imports, Exports, Speicher, Ressourcen

# LSTM

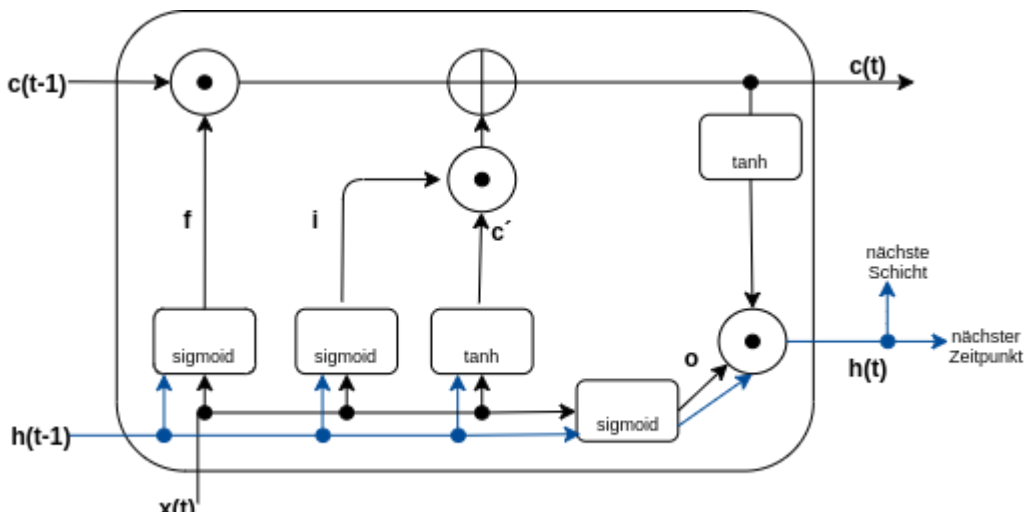


*Abb. 6: Struktur einer LSTM Zelle*
*(eigene Darstellung in Anlehnung an Raschka und Mirjalili (2019))*

# Leistungsmetriken

## Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

*Abb. 7: Accuracy Formel*
*(eigene Darstellung)*

## loss

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij})$$

*Abb. 8: loss Formel*
*(eigene Darstellung)*