Institut für Wirtschaftsinformatik
Humboldt-Universität zu Berlin

Identification of Cyber Attacks using Machine Learning

Georg Velev
Iliyana Pekova

velevgeo@hu-berlin.de
pekovail@hu-berlin.de

Agenda: Data Analysis and Data Corrections

1. Introduction: Dataset & theoretical Model
2. Literature Search
3. Duplicated Observations
4. Missing Values
5. Distribution of the Target Variable
6. Average Duration
7. Distribution protocol_type
8. Handle categorical Data
9. Redundant Features: Correlation Matrix
10. Outlook: To-Dos

1. Introduction: Dataset & theoretical Model

- **Dataset:**
  - collected from a local area network (LAN) that operated as an actual US. military network
  - collection period: 9 weeks
  - in this research: 10% of the whole data used for the analysis

- **Dataset Summary:**
  - 42 columns: including the target variable
  *the 41 features divided into four groups:
  basic, content, time-based and traffic-based attributes.

1. Introduction: Dataset & theoretical Model

- **Theoretical Model:**
  Cross Industry Standard Process for Data Mining

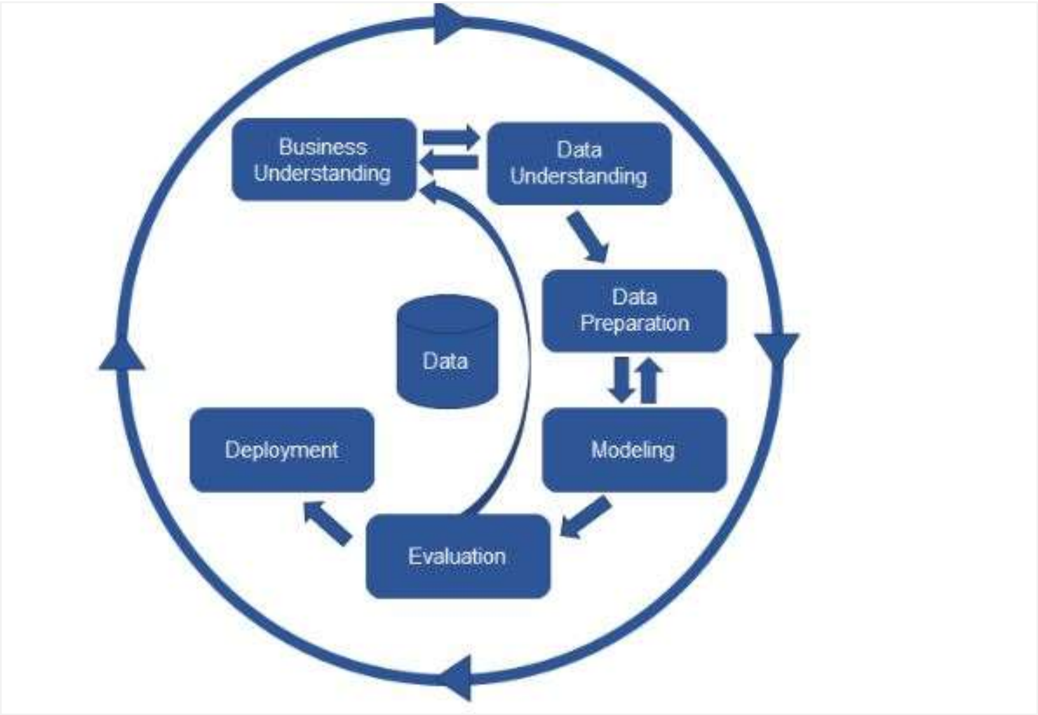

Figure 1: CRISP-DM (Taylor 2017)

2. Literature Search

Dataset	Machine Learning problem	Machine Learning Algorithms	source
KDD-Cup 1999	Anomaly Detection	Bayesian Filter	(Altwaijry and Algarny 2012)
		High Resolution Self-Organizing Map	(Saraswati et al. 2016)
	Classification Problem Anomaly Detection	Ant Colony Clustering	(Abdurrazaq et al. 2015)
NSL-KDD	Classification Problem	Random Forest	(Farnaaz and Jabbar 2016)
		K-means clustering	(Duque and Omar 2015)
		J48 Decision Tree, SVM, Naive Bayes	(Dhanabal and Shantharajah 2015)
NSL-KDD, ISCXIDS2012	Anomaly Detection	AdaBoost	(Mazini et al. 2018)
NSL-KDD, Kyoto 2006+		Discriminant Function Regularization Iterative Anomlay Repartition	(Aissa and Guerroumi 2016)
CIDDS-001	Classification Problem	Distance-based Machine Learning	(Verma and Ranga 2018)
UNSW NB-15	Classification Problem Anomaly Detection	Naive Bayes, REP Tree, Random Tree, Random Forest, Random Committee, Bagging, Randomizable Filtered Classifier	(Dahiya and Srivastava 2018)

Table 1: Literature Search

3. Duplicated Observations

- **Total amount of observations initially:** 494.020
- **71% of the data:** duplicates
- **Remaining only distinct observations:** 145.585

4. Missing Values

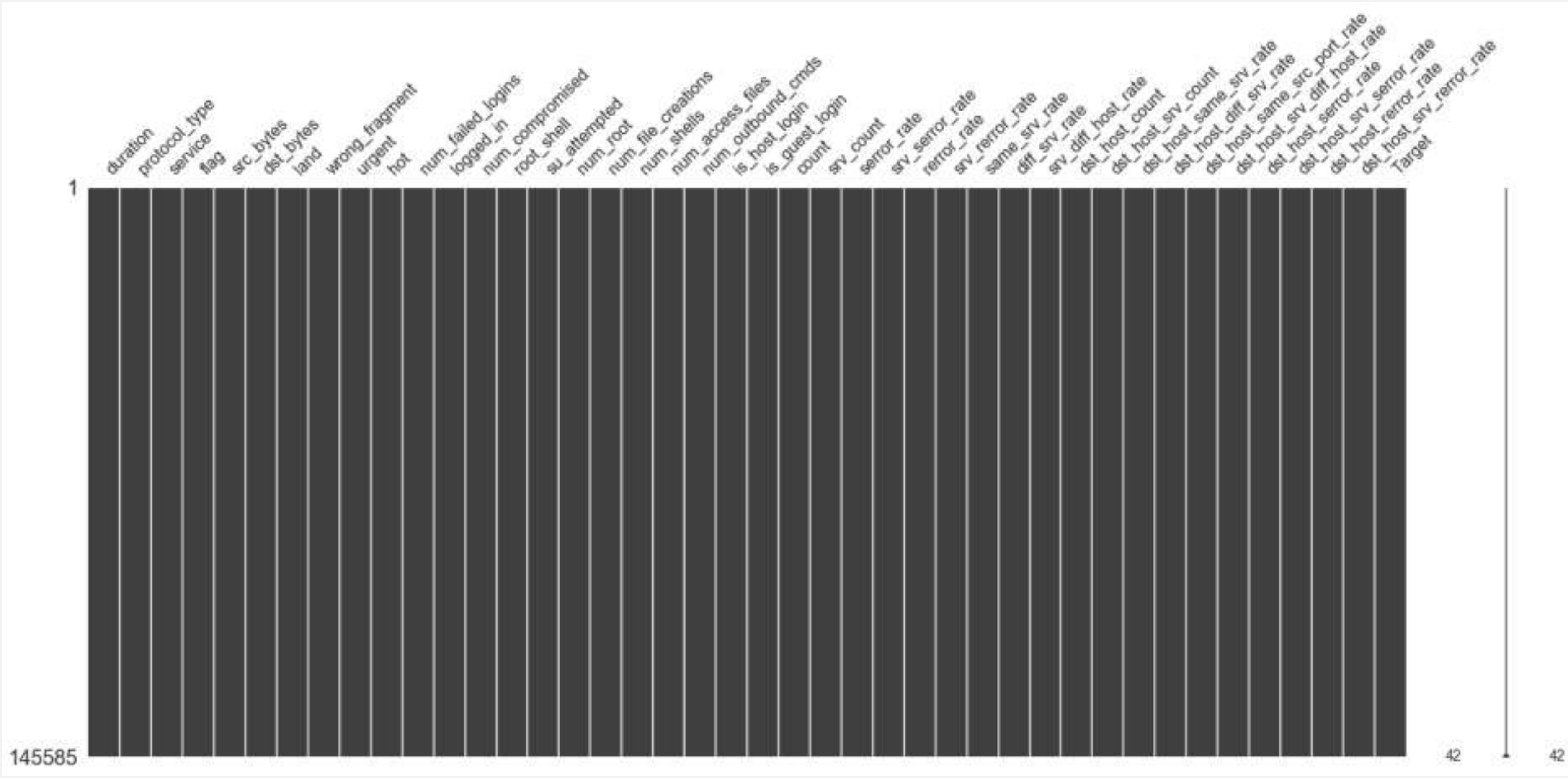

Figure 2: Missing Values KDD Cup Dataset

4. Missing Values

- **Benchmark:**

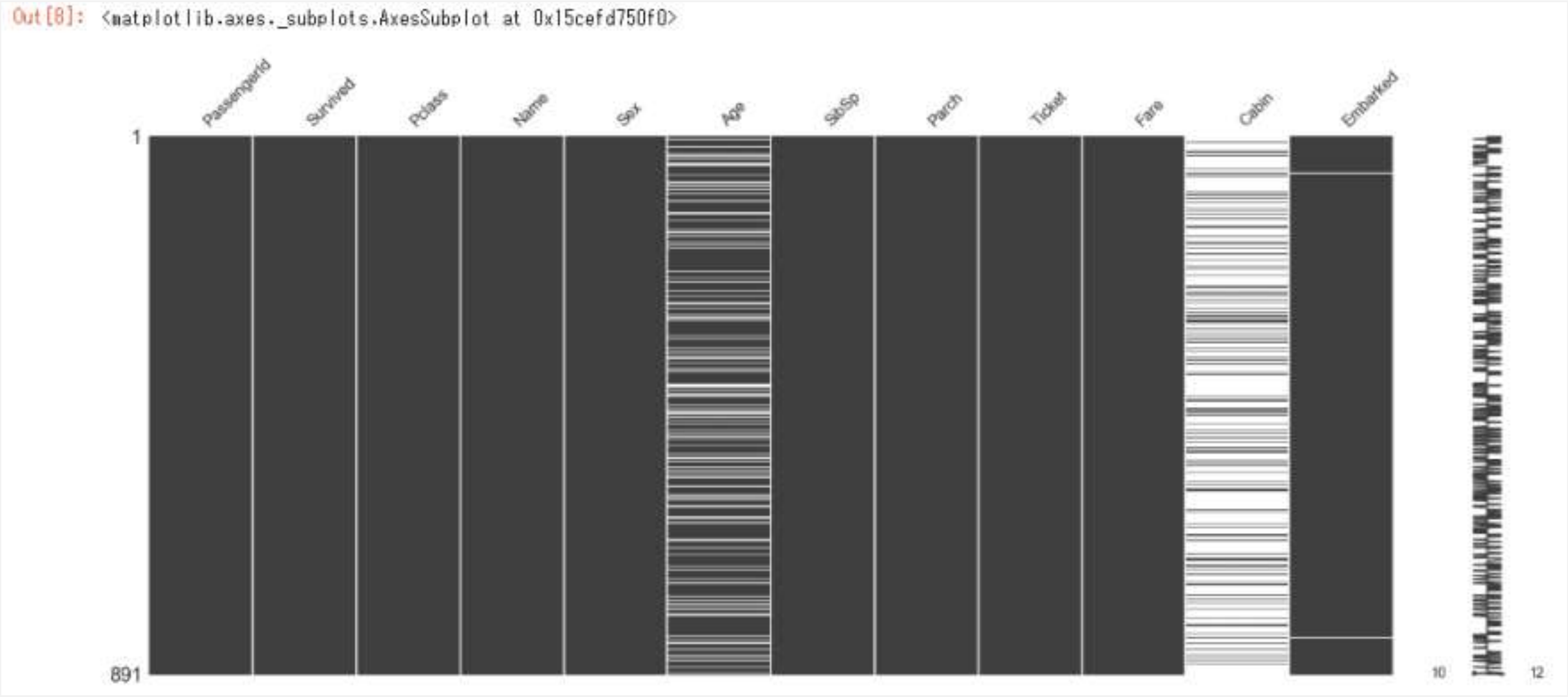

Figure 3: Missings Benchmark (Aota 2019)

5. Distribution of the Target Variable

- **Total of 24 cyber attacks:**

```
In [12]: print(values)
normal.              87831
neptune.             51820
back.                  968
teardrop.              918
satan.                 906
warezclient.           893
ipsweep.               651
smurf.                 641
portsweep.             416
pod.                   206
nmap.                  158
guess_passwd.           53
buffer_overflow.        30
warezmaster.            20
land.                   19
imap.                   12
rootkit.                10
loadmodule.              9
ftp_write.               8
multihop.                7
phf.                     4
perl.                    3
spy.                     2
```

Figure 4: Categories Target Variable

5. Distribution of the Target Variable

- **Divided in four main categories:**

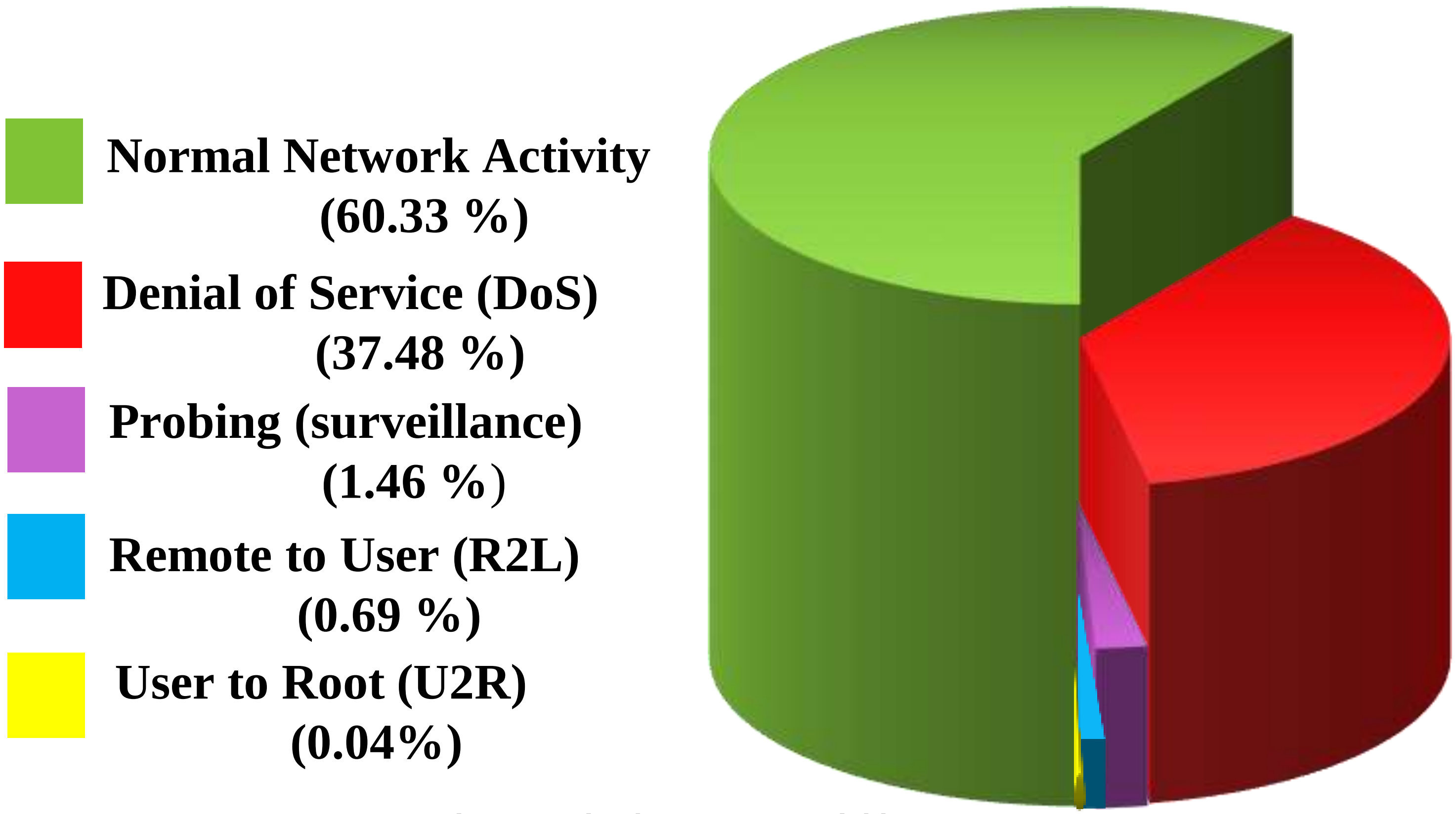

Figure 5: Pie Chart Target Variable

6. Average Duration

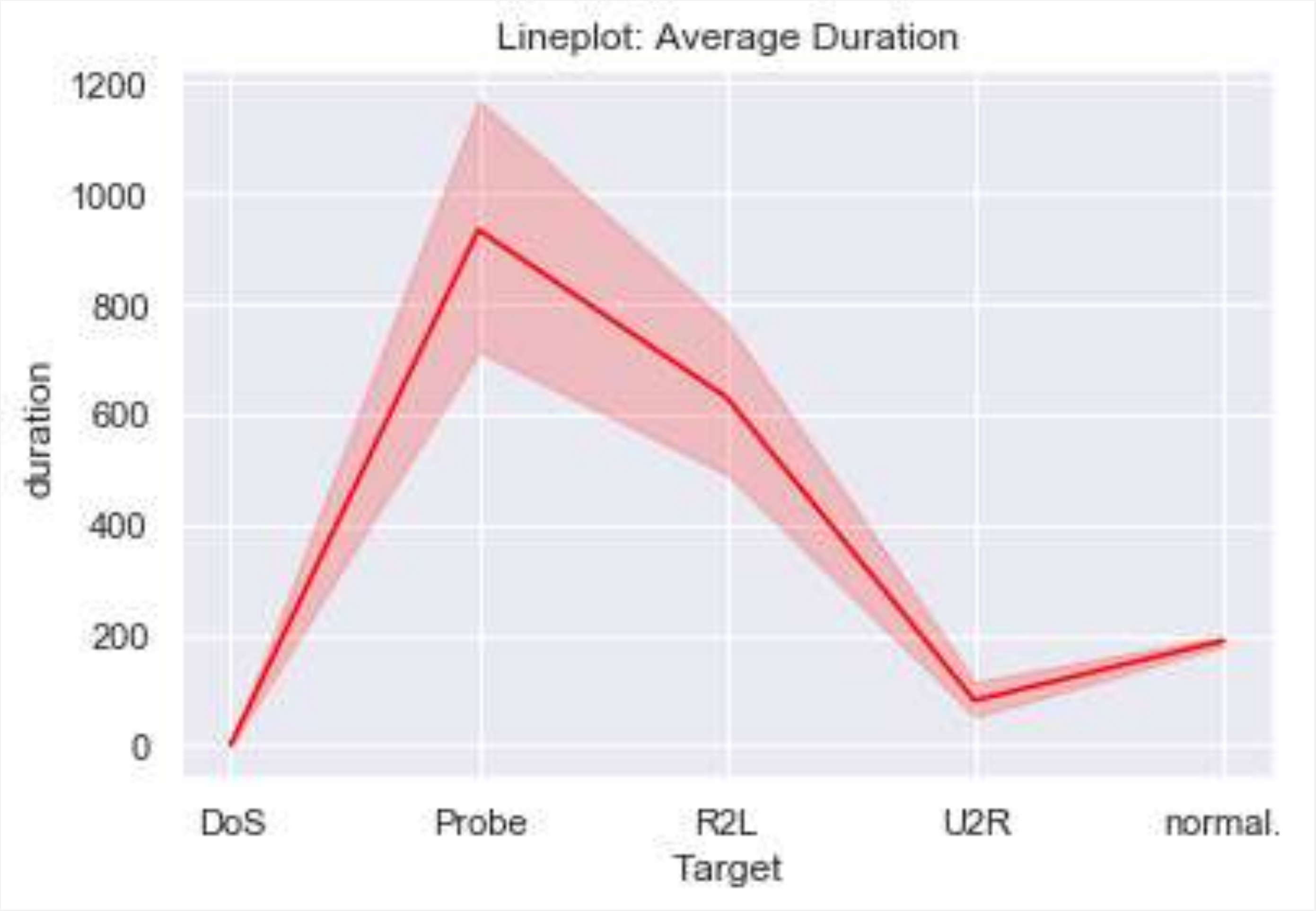

Figure 6: Line Plot average Duration

7. Distribution protocol_type

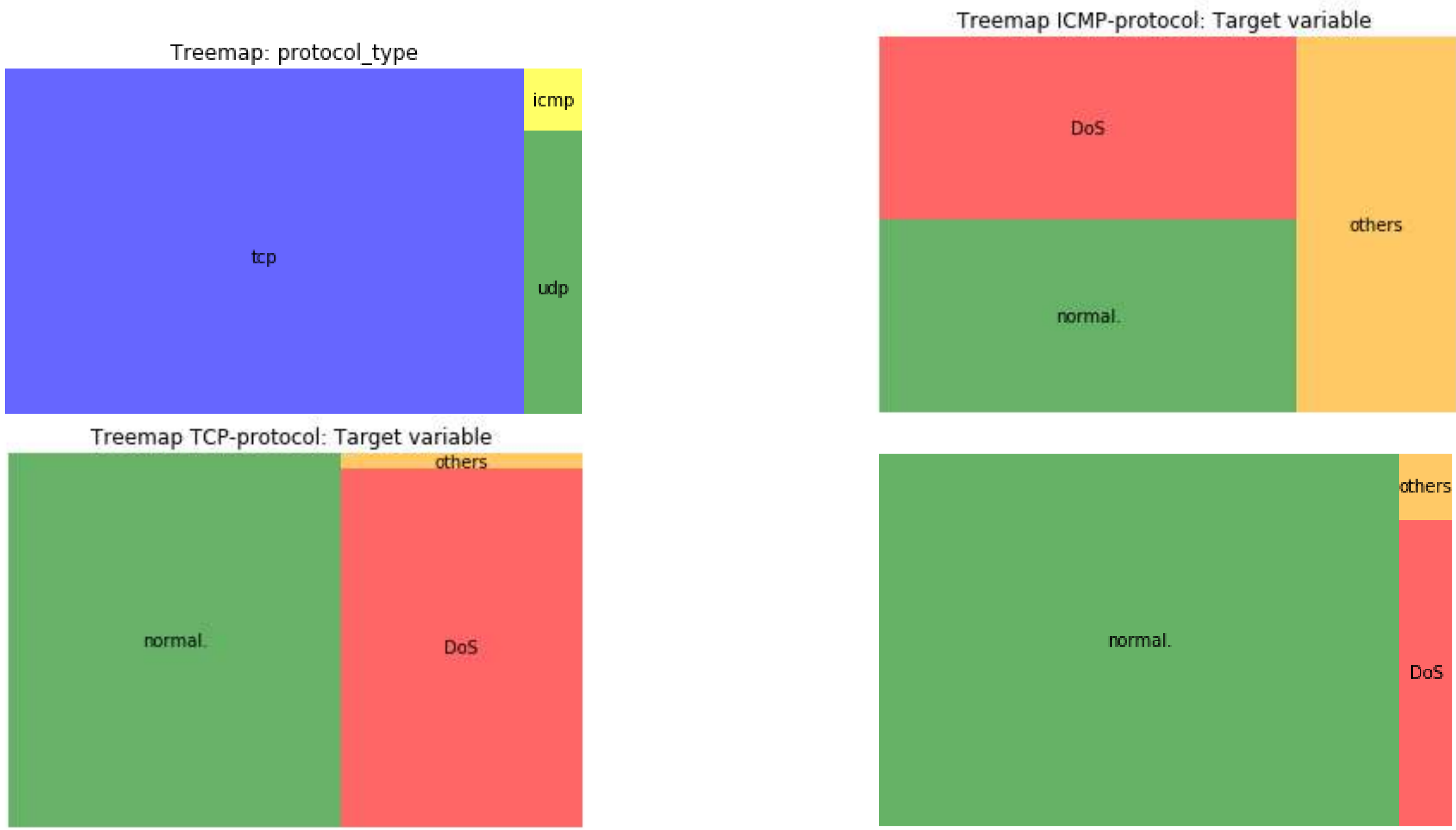

Figure 7: Treemaps protocol_type

8. Handle categorical Data

- **protocol_type:** most frequent category tcp
- **Stripplot:** protocol_type, Target, serror_rate

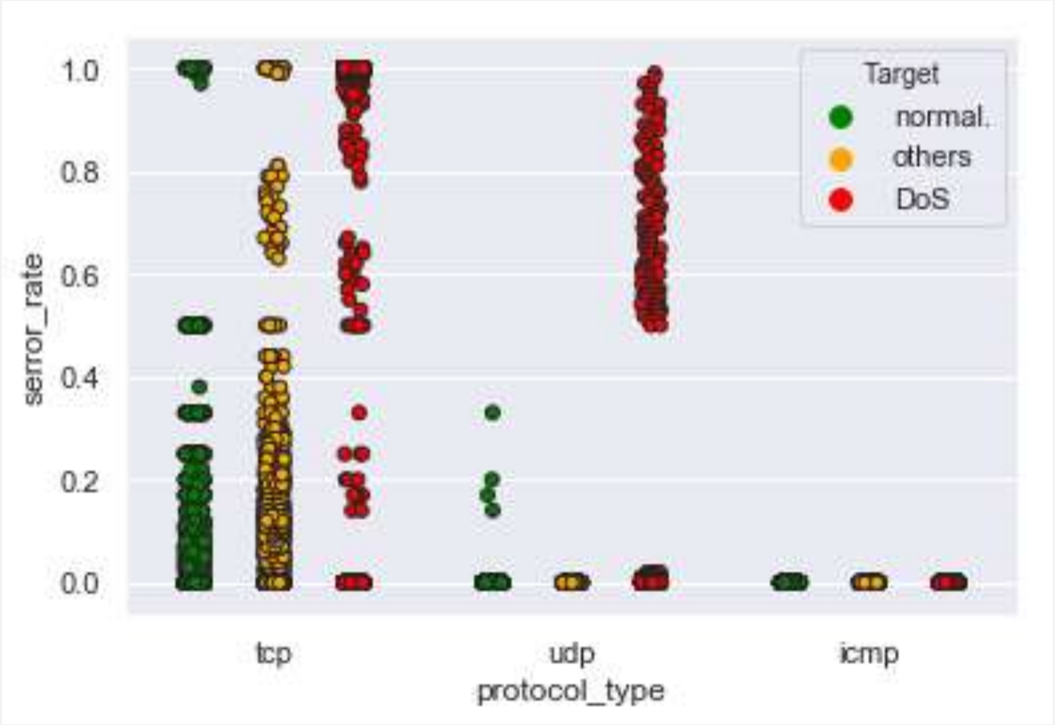

Figure 8: Stripplot TCP and SYN-Error

8. Handle categorical data

- **Countplot:** service

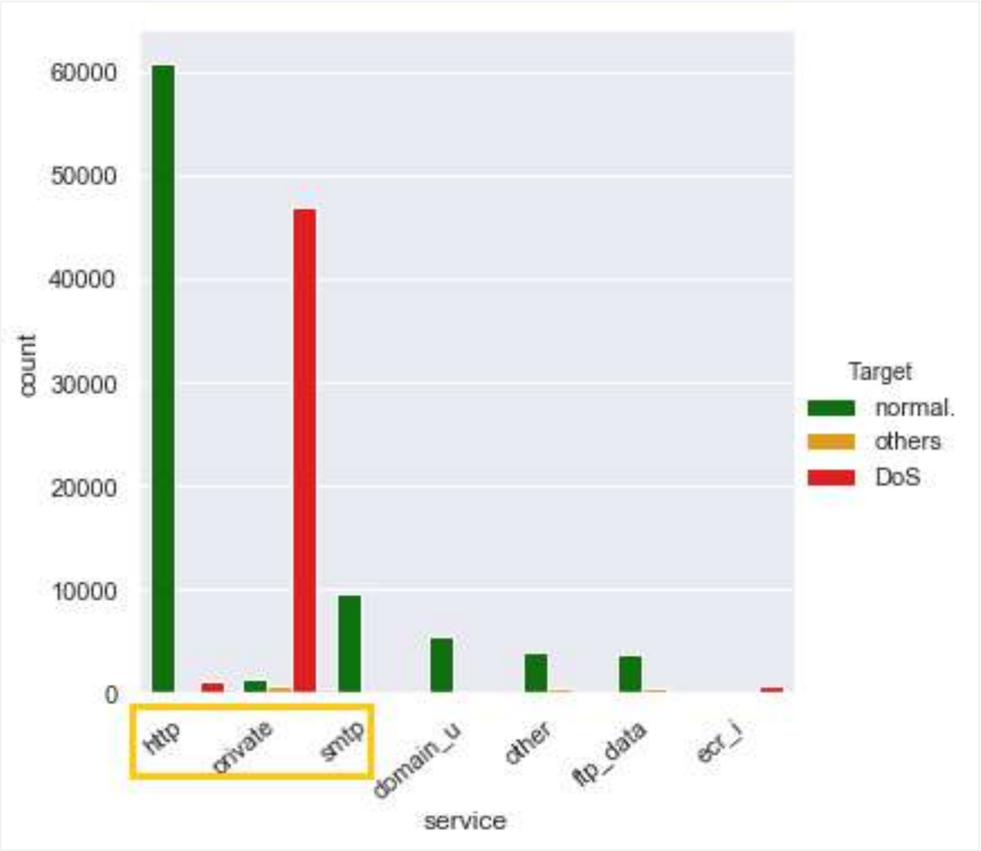

Figure 9: Countplot service

8. Handle categorical data

- **Countplot:** flag

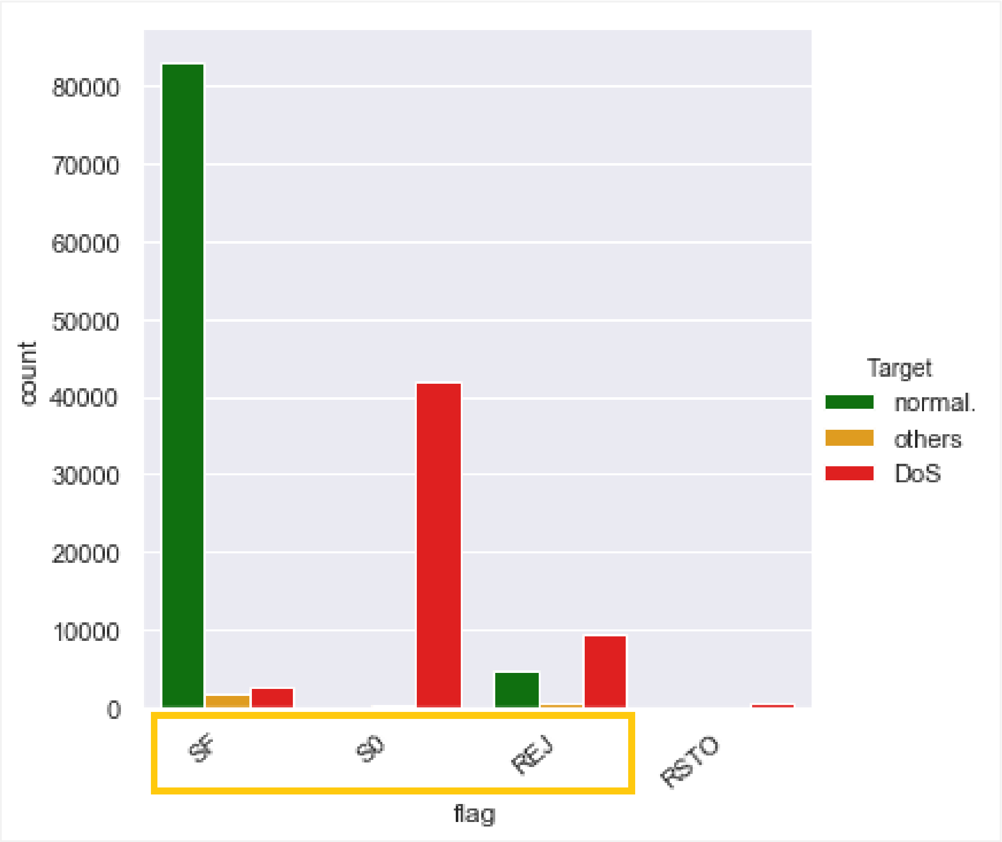

Figure 10: Countplot flag

9. Redundant Features

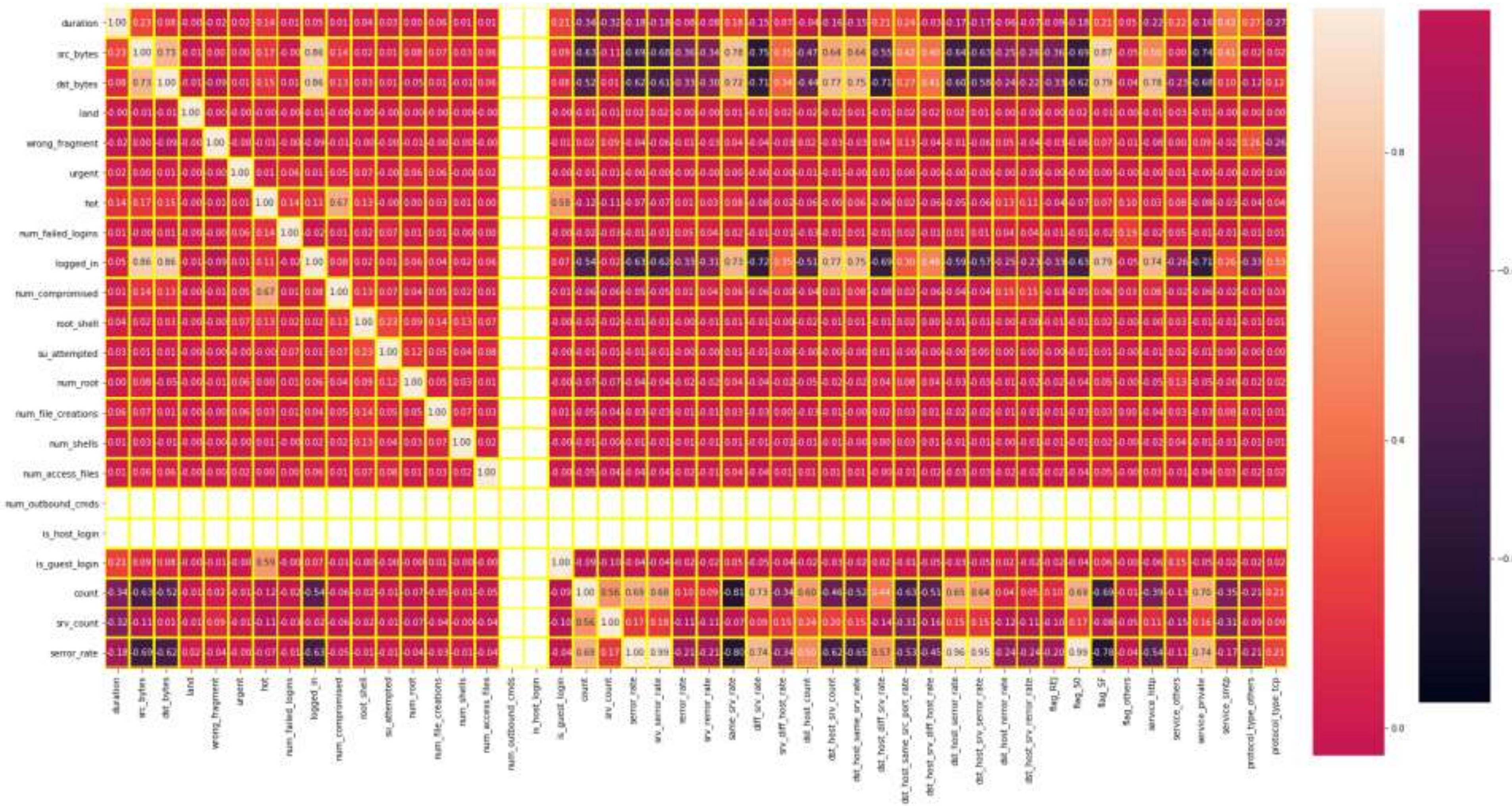

Figure 11: Heatmap Correlation Matrix

9. Redundant Features

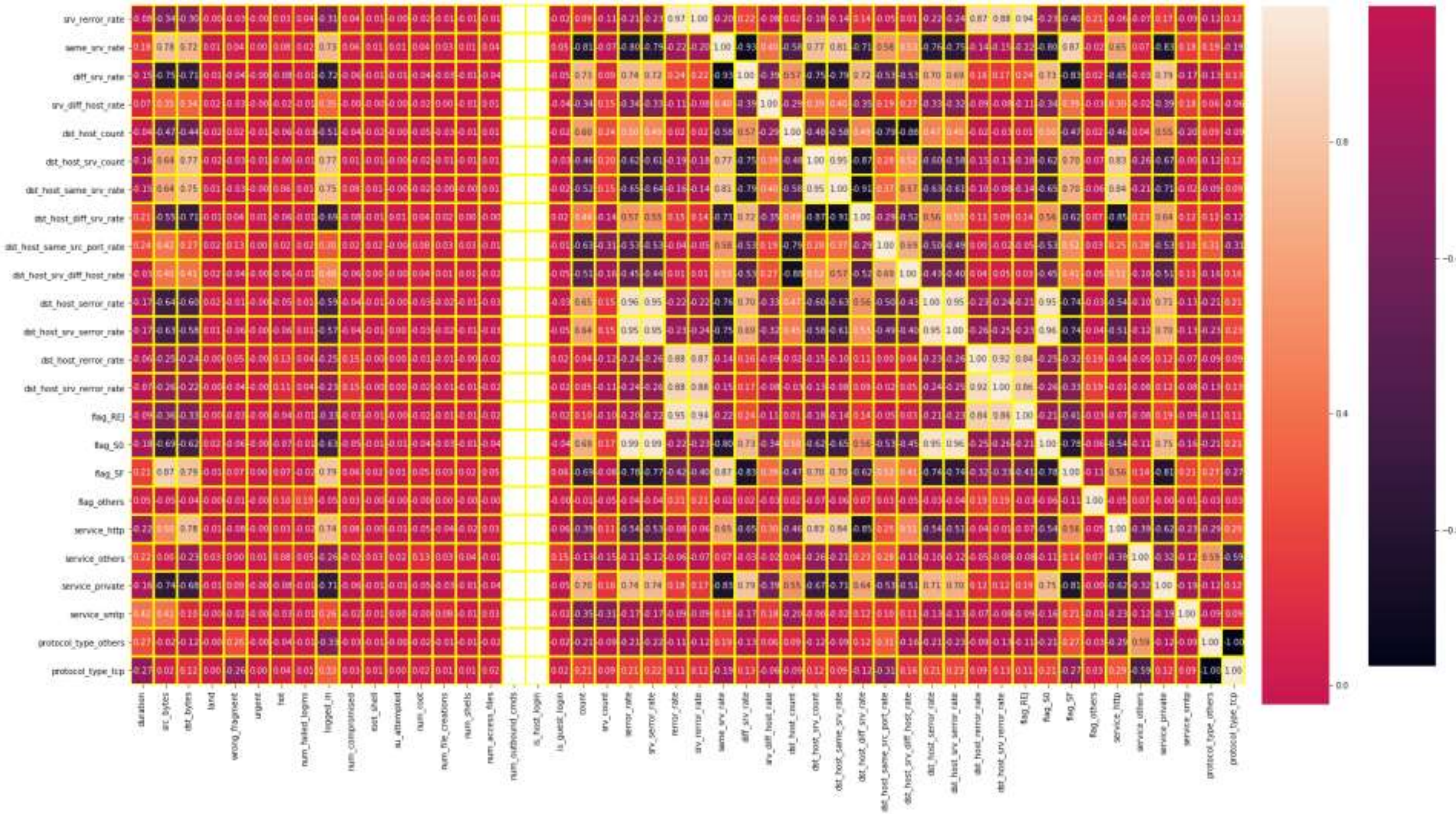

Figure 11: Heatmap Correlation Matrix

9. Redundant Features

Results for Naïve Bayes WITHOUT Binarization:

	Correlation-threshold	Detection Rate	Precision	False Negative Rate	AUC	False Alarm Rate
0	0.5	93.82%	93.74%	6.18%	94.86%	4.11%
1	0.6	93.82%	93.74%	6.18%	94.86%	4.11%
2	0.7	93.86%	93.69%	6.14%	94.86%	4.15%
3	0.8	93.77%	93.70%	6.23%	94.81%	4.14%
4	0.9	93.88%	99.38%	6.12%	96.75%	0.39%

Figure 12: Naïve Bayes Correlation Filter without Binarization

Results for Naïve Bayes WITH Binarization:

	Correlation-threshold	Detection Rate	Precision	False Negative Rate	AUC	False Alarm Rate
0	0.5	94.02%	99.73%	5.98%	96.93%	0.17%
1	0.6	94.04%	99.73%	5.96%	96.93%	0.17%
2	0.7	94.66%	99.82%	5.34%	97.27%	0.11%
3	0.8	93.71%	99.88%	6.29%	96.82%	0.07%
4	0.9	93.91%	99.77%	6.09%	96.89%	0.14%

Figure 13: Naïve Bayes Correlation Filter with Binarization

9. Redundant Features

Results for Random Forest:

	Correlation-threshold	Detection Rate	Precision	False Negative Rate	AUC	False Alarm Rate
0	0.5	99.83%	99.86%	0.17%	99.87%	0.09%
1	0.6	99.83%	99.90%	0.17%	99.88%	0.06%
2	0.7	99.85%	99.90%	0.15%	99.89%	0.07%
3	0.8	99.87%	99.96%	0.13%	99.92%	0.03%
4	0.9	99.85%	99.94%	0.15%	99.91%	0.04%

Figure 14: Random Forest Correlation Filter

10. Outlook: To-Dos

- **Extend the Literature Search**
- **Modelling:** Hyperparameter Optimization (Grid Search)

Reference List

Abdurrazaq, M. N. K., Trilaksono, B. R. & Rahardjo, B. (2015). DIDS Using Cooperative Agents Based on Ant Colony Clustering. *Journal of ICT Research and Applications 8* (3), 213-233.

Aissa, N. B. & Guerroumi, M. (2016). Semi-supervised Statistical Approach for Network Anomaly Detection. *Procedia Computer Science 83*, 1090–1095.

Altwaijry, H. & Algarny, S. (2012). Bayesian based intrusion detection system. *Journal of King Saud University - Computer and Information Sciences 24* (1), 1–6.

Aota, T. (2018). Visualizing the patterns of missing value occurrence with Python. https://dev.to/tomoyukiaota/visualizing-the-patterns-of-missing-value-occurrence-with-python-46dj. Accessed on: 03.02.2019.

Dahiya, P. & Srivastava, D. K. (2018). Network Intrusion Detection in Big Dataset Using Spark. *Procedia Computer Science 132*, 253–262.

Dhanabal, L. & Shantharajah, S. P. (2015). A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms. *International Journal of Advanced Research in Computer and Communication Engineering 4* (6), 446–552.

Duque, S. & Omar, M. N. b. (2015). Using Data Mining Algorithms for Developing a Model for Intrusion Detection System (IDS). *Procedia Computer Science 61*, 46–51.

Reference List

Institut für Wirtschaftsinformatik
Humboldt-Universität zu Berlin

Farnaaz, N. & Jabbar, M. A. (2016). Random Forest Modeling for Network Intrusion Detection System. *Procedia Computer Science 89*, 213–217.

Mazini, M., Shirazi, B. & Mahdavi, I. (2018). Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and AdaBoost algorithms. *Journal of King Saud University - Computer and Information Sciences.*

Saraswati, A., Hagenbuchner, M. & Zhou, Z. Q. (2016). High Resolution SOM Approach to Improving Anomaly Detection in Intrusion Detection Systems. In B. H. Kang & Q. Bai (Ed.), *AI 2016: Advances in Artificial Intelligence* (Lecture Notes in Computer Science, 191–199). Cham: Springer International Publishing.

Taylor, J. (2017). Four Problems in Using CRISP-DM and How To Fix Them. https://www.kdnuggets.com/2017/01/four-problems-crisp-dm-fix.html. Accessed on: 03.02.2019.

Verma, A. & Ranga, V. (2018). Statistical analysis of CIDDS-001 dataset for Network Intrusion Detection Systems using Distance-based Machine Learning. *Procedia Computer Science 125*, 709–716.