DOCTORAL SCHOOL OF BUSINESS INFORMATICS

THESIS SUMMARY

Csaba Brunner

INTRUSION DETECTION BY MACHINE LEARNING

SUPERVISOR:

Andrea Kő, Ph.D.

professor

Szabina Fodor, Ph.D.

associate professor

BUDAPEST, 2020

INSTITUTE OF INFORMATION TECHNOLOGY

THESIS SUMMARY

Csaba Brunner

INTRUSION DETECTION BY MACHINE LEARNING

SUPERVISOR:

Andrea Kő Ph.D.

professor

Szabina Fodor Ph.D.

associate professor

TABLE OF CONTENTS

1. MOTIVATION AND RESEARCH GOALS

The need to protect information systems and resources from misuse had arisen as early as 1972 and 1980, when James P. Anderson outlined that the USAF had become increasingly aware of information security related issues (Anderson, 1972, 1980). Since then, the reported number of system intrusions grew at an alarming rate, especially from the early 2000s, which, according to reports like (Beek *et al.*, 2019) only increased in severity. At the creation of this dissertation, the most common cyber attacks were the following:

- DDoS in the early 2000s (Lau *et al.*, 2000; Smith, 2014), causing significant revenue loss by shutting down services,
- Botnet infections in relation to DDoS (Smith, 2014), taking computational resources from legitimate clients and using those resources for illegal conduct,
- ransomwares, specialized malwares (Beek *et al.*, 2019), encrypting information and demanding ransom for decryption,
- and more recently, deepfake attacks (Damiani, 2019; Statt, 2019), where deep learning models are used to impersonate stakeholders in key positions to gain access to sensitive information or to conduct fraud.

The presence of these attacks changes among economic sectors, the most targeted being financial services, healthcare and education. Several methods exist for countering these malicious activities at different layers of an information system, a concept often referred to as defense in depth. One example is machine learning. Intrusive activities have well-defined patterns, detecting them is simple enough for a specialized system supported by the same machine learning algorithms. Furthermore, in some cases, like deepfake attacks, machine learning might be the only effective method of detection.

1.1. RESEARCH GOALS

Despite all this, machine learning techniques are still not widespread and utilized enough in IT security. This provided the motivation for studying network intrusion detection systems (NIDS) from a data mining perspective. The main goal was to provide a novel intrusion detection solution applying machine learning methods. This has been broken down to two additional goals:

RG1. To create an intrusion detection model that can compete with the ones introduced in related scientific literature, measured by detection performance metrics. Performance in this context is described as the portion of attacks correctly and incorrectly classified as being part of normal activity and vice versa. Several metrics are known to measure this performance, for example, accuracy and recall.

RG2. To identify machine learning methods that can improve performance on complex event detection problems where target features have a high degree of class imbalance. Intrusion detection fits this description, as the available data is heavily skewed towards the more common normal traffic, rather than the rarer malicious activity.

1.2. RESEARCH QUESTIONS

Based on the research goals the following research questions were formulated:

RQ1. Is machine learning a suitable approach for intrusion detection? If machine learning is a proper technique for intrusion detection, which are the appropriate models?

RQ2. Which type of intrusion detection method is more effective from the following ones: misuse detection by classification, anomaly detection by outlier analysis or a combination of the previous ones?

RQ3. What is the level of model performance that can be expected in an intrusion detection task?

RQ1 is interested in finding the right machine learning model, which is a challenging task. It is affected by the selected intrusion detection method (signature detection or anomaly detection) as well as the available dataset and the sampling method chosen for that dataset.

The most common and best working non-ensemble machine learning algorithms in intrusion detection are decision trees, artificial neural networks and k-nearest neighbor algorithms for signature detection. Each has drawbacks though:

- Decision trees are prone to overfitting, unstable (a small change in training data can cause entirely different decision trees) and perform poorly on unevenly distributed training classes.

- Artificial neural networks, like decision trees, are prone to overfitting, and generally have long training times.
- K-nearest neighbor algorithms are fast to train, but need all data for accurate predictions, therefore they scale poorly.

Although important, predictive performance is not the only characteristic for intrusion detectors to be compared by. Training time, prediction time and model portability are three additional characteristics to consider. However, the evaluation of these aspects was out of scope of this dissertation in favor of a more thorough study of predictions.

RQ2 is a more recent question in the field, highlighted by (Dua and Du, 2016), and is characterized by differences between the two key intrusion detection techniques, signature detection and anomaly detection. On one hand, signature detection can have high recall and low false positive rate, is easy to implement, and provides predictions quickly. However, it is incapable of detecting new and unknown attacks. On the other hand, anomaly detection aims at building a profile of normal traffic, and then detects anomalous or attack traffic based on the difference from this normal profile. Anomaly detection captures unknown attacks better; however, it is more difficult for it to set apart attacks and anomalous traffic, as the latter might include unusual, yet normal connections as well, therefore, anomaly detection has high false positive rates. Signature and anomaly detectors use compensatory detection approaches; therefore, it might be a good idea to combine them into new hybrid detectors.

A simple combination of the two techniques is not enough, a more purposeful approach must be followed. Good candidates for hybridization are models that do not perform conflicting operations on the data, for example, decision trees and one class SVM models or any autoencoder combined with fully connected artificial neural networks. The choice of integration can be simplified to one of the four alternatives (**Figure 1**), identified by (Dua and Du, 2016; Molina-Coronado *et al.*, 2020).

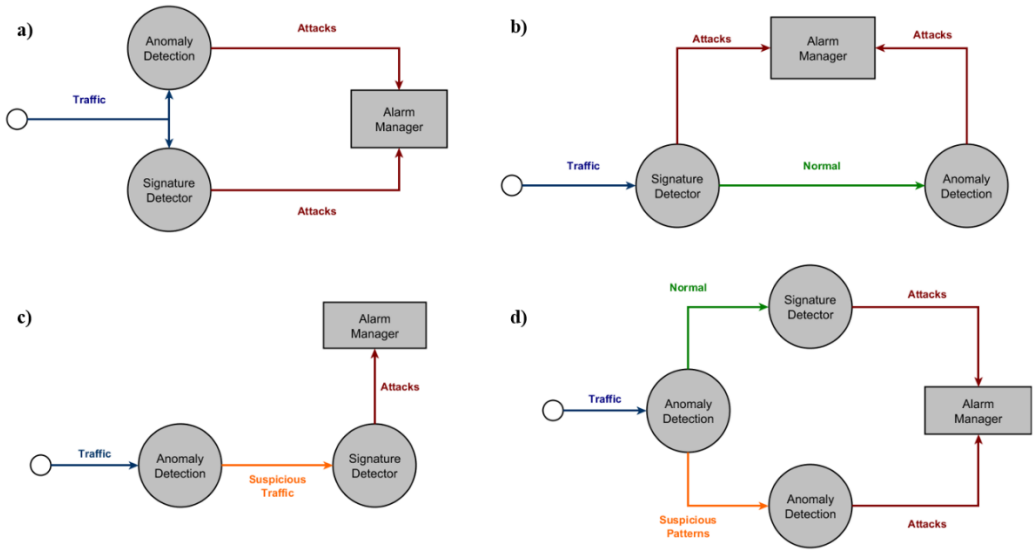

Figure 1: Types of hybrid intrusion detection. Source: (Molina-Coronado *et al.*, 2020)

The four identified potential hybrid intrusion detection technique:

- **Parallel detection**: used to correlate signature and anomaly detection results to provide a stronger detection (**Figure 1.a**). Network traffic is flagged as attack if either the anomaly or the signature detector identifies it as such.
- **Signature-Anomaly sequence detection**: designed to improve detection ability on unknown attacks missed by the signature detector (**Figure 1.b**).
- **Anomaly-Signature sequence detection**: designed to reduce false positive rates (**Figure 1.c**). The anomaly detector flags suspicious traffic, then the misuse detector confirms the flagged anomalies.
- **Complex mixture detection**: any detection approach using anomaly and signature detectors, that did not fit in the categories above (example: **Figure 1.d**).

Due to how the studied datasets were provided with dedicated target features, signature and hybrid detection approaches were available for evaluation.

Related to **RQ3** and based on reviewing the related literature, contemporary intrusion detection research is facing the following challenges:

- Predominant use of the accuracy measure for performance evaluation on data with unevenly distributed classes.
- Different articles created their own samples of a chosen dataset, making performance comparisons between them and the proposed models difficult, if not impossible.

- Intrusion detection is always involved with detecting minority classes.

There is a high variation on possible model performance measurements. Therefore, two criteria were set up for selecting papers from the related literature to compare the proposed models with, in order to test the assumptions of this dissertation.

- **Emphasis on recall / detection rate**: although accuracy is the most common metric, it is inappropriate for performing detections on imbalanced data. A better alternative is recall. Literature with recall as the model performance indicator are favored to those with accuracy, though, due to how common it is, accuracy could not be ignored completely.
- **Data sampling** is the second source of complexity and prediction variance in the literature. Different samples result in different models with different performance measurements. Therefore, only papers that validated their model proposals with the complete test samples of the datasets were used. Similarly, the intrusion detectors of this dissertation were set up in the following way: they were tested on the complete test portion of the respective dataset, regardless of the data used for training. This covered data preprocessing as well: transformations were performed on the test data using calculations from the training data to avoid information leakage.

In addition to comparisons with the related literature, **RQ3** also reflected on the potential techniques improving intrusion detection performance. These techniques included:

- **Synthetic sampling**: creates artificial samples from minority classes, increasing their weight within the dataset. Techniques included are synthetic minority oversampling technique (SMOTE - (Chawla *et al.*, 2002)) and its variations (SVM SMOTE - (Nguyen, Cooper and Kamei, 2009), SMOTE Tomek and SMOTE ENN - (Batista, Prati and Monard, 2004)).
- **Advanced hyperparameter optimization**: hyperparameter optimization is the automated choice of machine learning model parameters that are not optimized by the model itself. Simple techniques for hyperparameter optimization exist, although the dissertation demonstrated more advanced Bayesian optimization with Gaussian Process (GP) priors (Brochu, Cora and De Freitas, 2010; Snoek, Larochelle and Adams, 2012) and used tree-structured parzen estimators (TPE -

(Bergstra *et al.*, 2011; Bergstra, Yamins and Cox, 2013)) to improve detection performances further.

- **Ensemble models**: techniques to combine individual base model predictions into an aggregate classifier to improve bias (boosting), reduce variance (bagging) or both (stacking)(Smolyakov, 2017; Budzik, 2019).

2. BACKGROUND

A definition for intrusions and intrusion detection: *"Any unauthorized attempt to access, manipulate, modify, or destroy information or to use a computer system remotely to spam, hack, or modify other computers. An IDS intelligently monitors activities that occur in a computing resource, e.g., network traffic and computer usage, to analyze the events and generate reactions"* (Dua and Du, 2016, p. 10). This description accounts for botnet activities and includes both network and host intrusion detection. A second definition for intrusion detection systems: "*Intrusion Detection Systems are deployed to uncover cyberattacks that may harm information systems.*" (Molina-Coronado *et al.*, 2020, p. 2). In this dissertation, IDS will be treated as a system designed to detect attempts at unauthorized access to an information system coming from a wider external network. The key assumption for such a system to function is that intrusive behavior is discernable from normal activity.

(Scarfone and Mell, 2007; Dua and Du, 2016; Molina-Coronado *et al.*, 2020) distinguished multiple types of intrusion detection systems based on what it aims to protect, the most important of which are:

- **Network based (NIDS)**: monitoring traffic on network devices or segments with the aim of detecting malicious traffic aimed at devices within the protected network boundaries. Network intrusion detectors are usually deployed in DMZs, as part of an intelligent firewall, VPN servers, remote access servers and wireless network access points.

- **Host based (HIDS)**: monitoring the resource consumption on a single system for suspicious activity. This host can be a critical IT infrastructure element, typically an application or database server.

Data mining and machine learning are just two of the many techniques used for intrusion detection. From a data mining perspective, (Scarfone and Mell, 2007; Dua and Du, 2016; Molina-Coronado *et al.*, 2020) distinguished IDS systems further into:

- **Misuse / signature detection**: IDS that generates alarms when a known intrusion occurs. Known attacks can be detected reliably with low false positive rates, however new attacks cannot be detected. Misuse detectors describe known attacks

as malicious patterns; therefore, they require data on the attacks first to be able to detect them.

- **Anomaly-based detection**: alarms are triggered when a traffic flow behaves in a significantly different way compared to normal traffic patterns. Subsequently, they can detect previously unknown attacks at the cost of a higher false positive rate.

- **Hybrid detection:** to improve the detection performance of IDSs, some researchers proposed to combine anomaly and misuse detection into hybrid detectors. The underlying idea is to combine the benefits of the two, like the ability to detect known attacks with low false positive rates, while maintaining some ability of detecting new attacks when needed.

Data mining has several definitions, (Fayyad, Piatetsky-Shapiro and Smyth, 1996) defined it as a part of a wider process called knowledge discovery in databases (KDD). KDD is determined as *"the overall process of discovering useful knowledge from data"* (Fayyad, Piatetsky-Shapiro and Smyth, 1996, p. 40) and data mining as *"a process using statistical, mathematical and artificial intelligence techniques to extract and identify useful information and subsequent knowledge from large sets of data"*. From the perspective of an IDS, the hidden knowledge is the unknown intent behind the source of the network traffic and the data is the inbound network traffic. The goal is to set apart traffic sent with malicious intent from the legitimate.

The terms data mining and machine learning, depending on interpretation, are often used as synonyms. This dissertation will use the following definition for machine learning: "*it is a field of study that gives computers the ability to learn without being explicitly programmed to*" (Samuel, 1959, indirect quote). Data mining is an activity in the KDD process, producing patterns to discover interesting knowledge. Machine learning algorithms are frequently, though not exclusively, used in data mining to generate these patterns.

3. METHODOLOGY

The methodology used for designing, executing and evaluating the proposed models follow a top-down pattern (**Figure 2**). The research goals of this dissertation can be achieved by creating and evaluating an algorithmic artifact. The goal of design science research is to answer research questions with design artefacts; therefore, it was found to be a fitting methodology, formulating the first methodological pillar of the study. In addition, the algorithmic artifact is a machine learning model, therefore the concepts and considerations of the CRISP-DM process model for planning, implementing and deploying machine learning models apply as well, forming the second methodological pillar. Finally, the proposed model designs outline the exact process of model creation, with the necessary data preprocessing, training and evaluation steps involved, forming the third and lowest level of methodological abstraction.

Figure 2: The methodological abstraction levels followed in this dissertation. Source: own edit.

3.1. DESIGN SCIENCE RESEARCH AND CRISP-DM

"*Design science is the design and investigation of artifacts in context. The artifacts we study are designed to interact with a problem context in order to improve something in that context.*" (Wieringa, 2014, p. 3) It is the simultaneous study of research questions and the development of an IT artifact (**Figure 3**).

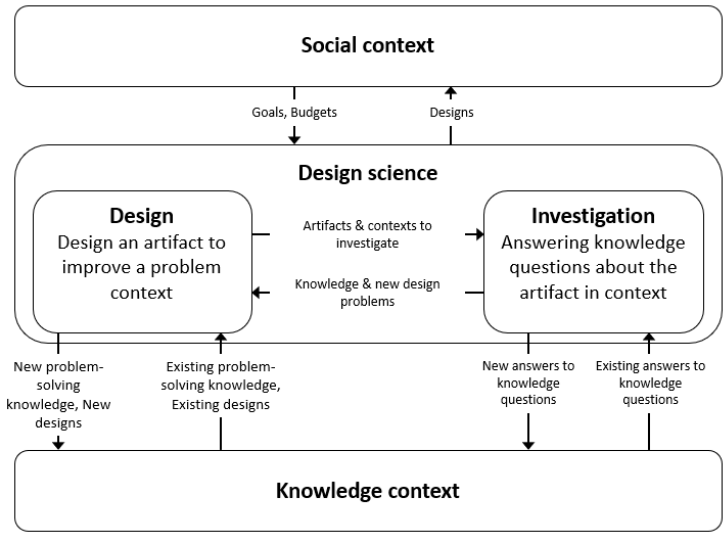

Figure 3: The design science framework. Source: (Wieringa, 2014)

Design science research is characterized by its social and knowledge contexts. The social context consists of stakeholders who define goals and requirements for and provide the budget of the research project, while expecting a tangible outcome (treatment) achieving these goals. On the other hand, the knowledge context is often used to address design and investigation challenges and is further enriched with new designs and answers to knowledge questions once the research project has concluded.

Design and engineering cycles form one of the pillars for design science research, responsible for the creation, evaluation, validation and implementation of artifacts addressing the goals. The engineering cycle is a rational problem-solving process to deliver a working treatment consisting of 5 steps (**Figure 5**). The design cycle summarizes the first three steps of the engineering cycle and aims to create a treatment design.

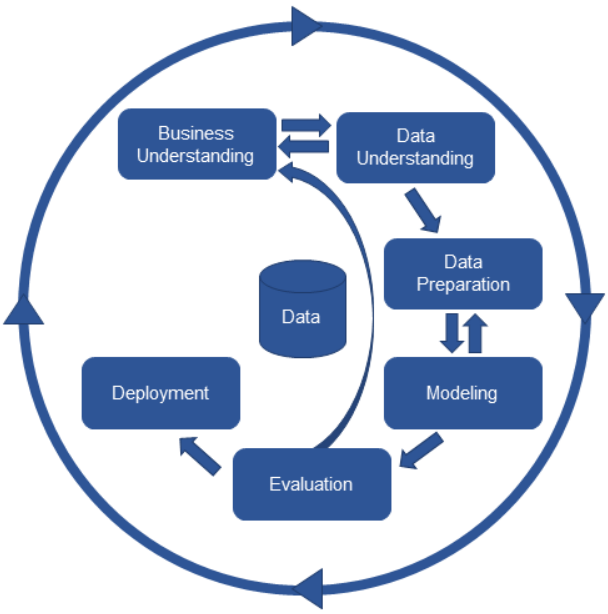

Figure 4: The CRISP-DM process model. Source: (Chapman *et al.*, 2000)

To systematically carry out data mining projects, a general process flow is required. One of the most popular data mining process models is the cross industry standard process for data mining (CRISP-DM) designed by (Chapman *et al.*, 2000). The CRISP-DM process (**Figure 4**) starts with a good understanding of the business and the associated need for data mining and ends with the deployment of a machine learning model that satisfies the specified business need.

The two main methodologies of the dissertation are connected by their logically corresponding tasks (**Figure 5**), for example, problem investigation in the engineering cycle involves activities that are similar to activities performed during the business understanding and data understanding tasks of CRISP-DM. As the design cycle is only interested in the first three steps of the engineering cycle, the dissertation only discussed CRISP-DM tasks leading up to and including model evaluation, leaving deployment activities out of scope. Furthermore, due to the circumstances of the dissertation, a detailed study of the social context was infeasible, therefore that too, was left out of scope.

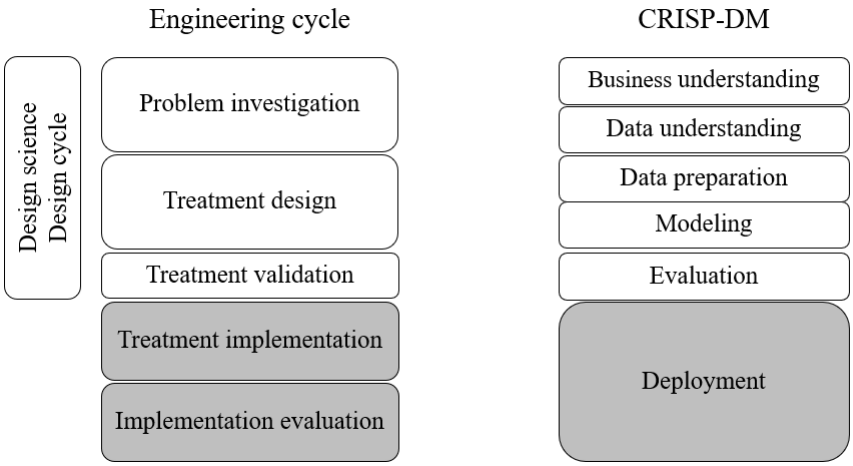

Figure 5: The relationship between the Engineering Cycle and CRISP-DM. Based on: (Chapman et al., 2000; Wieringa, 2014)

The two methodologies have differences as well. For example, the two have nuanced differences in their respective goals. The main goal of design science research is not only to deliver a well-designed, working artifact, but also to answer scientific questions about the artifact, its context or the relationship between the two. Comparatively, the goal of CRISP-DM is more practical. It is interested in delivering a machine learning algorithm, preferably as a part of a working business solution or service, delivering value to both customers and organization. The CRISP-DM approach therefore is more focused on evaluating the business context and the effects on the business context, rather than on answering research questions.

3.2. PROPOSED MODELS

As the third and final methodological pillar, the designs and evaluations of the proposed models were provided in iterative steps according to the CRISP-DM process model, broken down to data preprocessing, modeling, the evaluation of potential model

improvements and, in separate chapters, detection performance evaluation. The design of each model variant yielded a new, more refined version of an intrusion detector (**Figure 6**):

- **Version 0** (prototype): the first prototype of the model was outlined and evaluated in (Brunner, 2017), where a decision tree bagging classifier was published and trained on a map-reduce-like architecture.
- **Version 1** (neural network stacking ensemble): was a stacking ensemble built from neural networks trained on different features. Performance was improved by a more robust sampling process and grid search hyperparameter optimization.
- **Version 2** (migration to TensorFlow): The neural network ensemble was moved over to Keras on TensorFlow backend, achieving improved training. Further expected improvements in predictions were attempted by the use of TPE hyperparameter optimization. A second goal with this variant was to evaluate different variations of SMOTE sampling, namely SMOTE ENN, SMOTE Tomek, and SVM SMOTE.
- **Version 3** (extension with autoencoders): where the best performing elements of the earlier version (like SVM SMOTE sampling and TPE optimization) were extended with deep autoencoder networks trained on normal traffic, creating a hybrid intrusion detection approach.

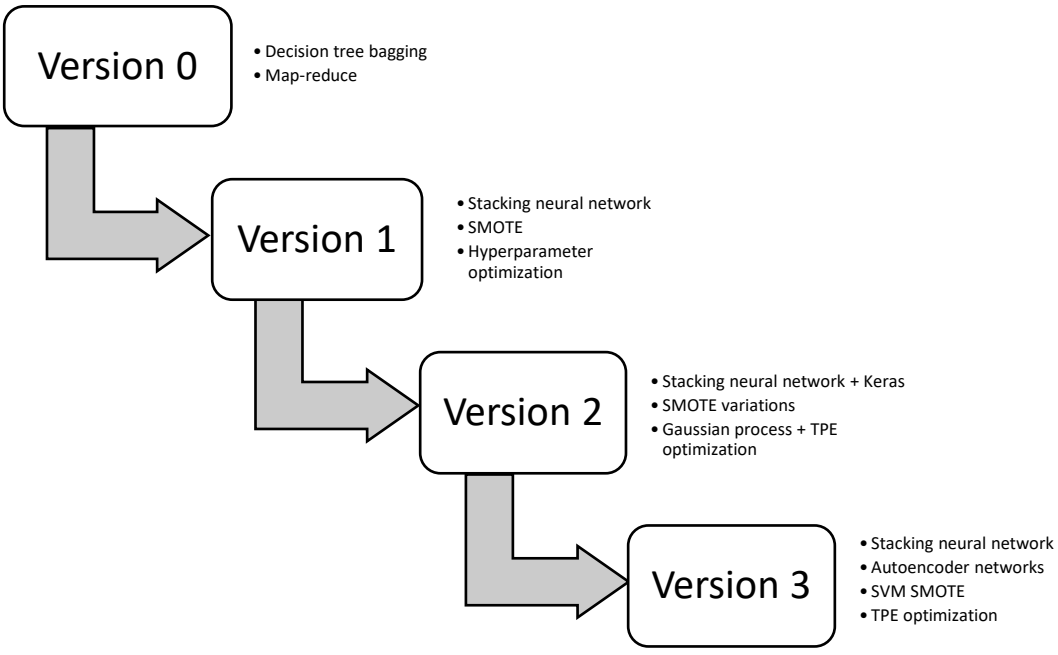

Figure 6: Iterations on the studied detection model. Source: own edit.

These intrusion detector variants were trained and evaluated on two related datasets, the KDD Cup 1999 (Stolfo *et al.*, 2000) and the NSL-KDD (Tavallaee *et al.*, 2009) both acting as benchmark datasets for intrusion detection model comparison. The difference between the two is that the NSL-KDD dataset fixed a data redundancy issue with the KDD Cup 1999, that caused earlier models to be biased towards repeated observations. Out of the proposed models, V0 was trained and evaluated on KDD Cup 1999, V1 on both, and the rest only on NSL-KDD. Though old, these datasets are still viable for detection benchmarks, due to how they model the appearance of new intrusion techniques with new attack types present in the test data only.

4. SUMMARY OF RESEARCH RESULTS

The results (**Table 1**, **Table 2** and **Table 3**) show the measured performances for each intrusion detector model variant (V1-3) compared to one another and a selected list of intrusion detector proposals found in the related literature. The key metrics for comparison were accuracy and recall. To address **RQ1**, machine learning proved to be a suitable approach for intrusion detection, however, its true effectiveness is more nuanced. Based on the related literature, a certain hierarchy can be set up between simple models, model ensembles, hybrid detectors and models enhanced with data generative solutions, like autoencoder networks in that order. This has been proven by the V1-3 models designed for the dissertation as well.

	V1	V2			V3
		SMOT ENN	SMOTE Tomek	SVM SMOTE	
Normal	0.9452	0.9255	0.9198	0.9140	0.8367
DoS	0.8126	0.8259	0.8592	0.8438	0.7728
Probe	0.6898	0.5225	0.5580	0.5944	0.7732
R2L	0.2400	0.3258	0.3289	0.3109	0.3262
U2R	0.1045	0.3731	0.2985	0.2985	0.5821
Average	0.5584	0.5946	0.5929	0.5923	0.6582

Table 1: Recall table for all experiments. Source: own edit

Addressing **RQ2**; based on recall measurements (**Table 1**), the V3 variant traded performance on majority classes (normal and DoS) for performance on minority classes, especially U2R. This caused a significant increase in the macro-averaged recall. For accuracy tables, the same does not hold true, V3 performed the worst (with 74.26%), whereas V2 SMOTE Tomek provided the best (with 78.34%) results. The choice of approach depends on the purpose of intrusion detection, where the misclassification of an attack as normal traffic has far-reaching consequences. These consequences are better captured by recall, therefore the application of V3 is preferred in practice.

Model	Accuracy	Recall
KNN (Yang *et al.*, 2019)	76.51%	48.3%
Multinomial NB (Yang *et al.*, 2019)	78.73%	47.69%
RF (Yang *et al.*, 2019)	76.49%	48.84%
SVM (Yang *et al.*, 2019)	72.28%	45.88%
DNN (Yang *et al.*, 2019)	80.22%	52.77%
DBN (Yang *et al.*, 2019)	80.82%	53.61%
ROS-DNN (Yang *et al.*, 2019)	78.26%	49.59%
SMOTE-DNN (Yang *et al.*, 2019)	81.16%	51.49%
ADASYN-DNN (Yang *et al.*, 2019)	80.1%	51.47%

Model	Accuracy	Recall
ICVAE-DNN (Yang *et al.*, 2019)	85.97%	62.66%
VGM + RF (Lopez-Martin, Carro and Sanchez-Esguevillas, 2019)	73.61%	N/A
VGM + Logistic Regression (Lopez-Martin, Carro and Sanchez-Esguevillas, 2019)	77.29%	N/A
VGM + Linear SVM (Lopez-Martin, Carro and Sanchez-Esguevillas, 2019)	77.23%	N/A
VGM + MLP (Lopez-Martin, Carro and Sanchez-Esguevillas, 2019)	79.26%	N/A
SVM SMOTE + RF (Lopez-Martin, Carro and Sanchez-Esguevillas, 2019)	74.25%	N/A
SVM SMOTE + Logistic Regression (Lopez-Martin, Carro and Sanchez-Esguevillas, 2019)	76.29%	N/A
SVM SMOTE + Linear SVM (Lopez-Martin, Carro and Sanchez-Esguevillas, 2019)	77.99%	N/A
SVM SMOTE + MLP (Lopez-Martin, Carro and Sanchez-Esguevillas, 2019)	77.98%	N/A
Decision Tree (Yin *et al.*, 2017)	74.6%	N/A
NB (Yin *et al.*, 2017)	74.4%	N/A
RF (Yin *et al.*, 2017)	72.8%	N/A
NB Tree (Yin *et al.*, 2017)	75.4%	N/A
MLP (Yin *et al.*, 2017)	78.1%	N/A
RNN (Yin *et al.*, 2017)	81.29%	N/A
SAE + SMR (Javaid *et al.*, 2016)	79.1%	N/A
AE + SVM (Al-Qatf *et al.*, 2018)	80.48%	N/A
Proposed V3 (AE + Stacking NN)	74.26%	65.82%
Proposed V2 + SMOTE ENN	77.09%	59.46%
Proposed V2 + SMOTE Tomek	78.34%	59.29%
Proposed V2 + SVM SMOTE	77.75%	59.23%
Proposed V1 (Stacking NN)	78.11%	55.84%

Table 2: External comparisons in terms of accuracy and recall. Source: own edit

Results collected from the related literature are from articles studying hybrid intrusion detection with autoencoder networks. These articles also included more simple models among the results, included in the comparison as well. Answering **RQ3**, all proposed model variants outperformed mean performance aggregated over the related intrusion detection literature (**Table 2**), and the V3 model achieved the best overall recall. Furthermore, (Yang *et al.*, 2019) provided per-class recall values as well, enabling a more detailed analysis (**Table 3**).

Model	Normal	DoS	Probe	R2L	U2R
KNN (Yang *et al.*, 2019)	92.78%	82.25%	59.4%	3.56%	3.5%
Multinomial NB (Yang *et al.*, 2019)	96.03%	37.1%	82.61%	22.22%	0.5%
RF (Yang *et al.*, 2019)	97.37%	80.24%	58.53%	7.55%	0.5%
SVM (Yang *et al.*, 2019)	92.82%	74.85%	61.71%	0%	0%
DNN (Yang *et al.*, 2019)	96.1%	85.4%	65.3%	14.56%	2.5%
DBN (Yang *et al.*, 2019)	97.04%	83.11%	69.85%	12.56%	5.5%

Model	Normal	DoS	Probe	R2L	U2R
ROS-DNN (Yang *et al.*, 2019)	92.61%	80.32%	56.26%	12.75%	6%
SMOTE-DNN (Yang *et al.*, 2019)	96.59%	82.19%	56.75%	10.93%	11%
ADASYN-DNN (Yang *et al.*, 2019)	96.43%	83.28%	59.81%	9.84%	8%
ICVAE-DNN (Yang *et al.*, 2019)	97.26%	85.65%	74.97%	44.41%	11%
Proposed V3 (AE + Stacking NN)	83.67%	77.28%	77.32%	32.62%	58.21%
Proposed V2 + SMOTE ENN	92.55%	82.59%	52.25%	32.58%	37.31%
Proposed V2 + SMOTE Tomek	91.98%	85.92%	55.80%	32.89%	29.85%
Proposed V2 + SVM SMOTE	91.40%	84.38%	59.44%	31.09%	29.85%
Proposed V1 (Stacking NN)	94.52%	81.26%	68.98%	24.00%	10.45%

Table 3: Recall comparison per class. Source: (Yang et al., 2019) & own edit

The mean recall values were 95.5% for normal, 77.44% for DoS, 64.52% for probe, 13.84% for R2L and 4.85% for U2R classes. The proposed models performed under average for normal classes, above average with the exception of V3 for DoS attacks, above average except for the V2 models for probe, and all proposed models performed above average for R2L and U2R classes. The autoencoder enhanced model proposal provided the worst recall on normal traffic and on DoS attacks compared to the measurements. The V3 model performed better, however, at predicting probe, U2R and R2L attacks. The V3 model traded good performance on majority classes for better classifications on minority classes.

In relation to **RQ3**, an additional goal was the identification techniques that could help machine learning models achieve increased intrusion detection performance. V2 models were set up to test multiple variants of SMOTE. Based on the achieved results (**Table 1**), no variant managed to significantly outperform the others, however the application of SMOTE did increase detection ability. An additional technique identified was advanced hyperparameter optimization by using TPE approach in all the V2 variants and in V3.

5. PUBLICATIONS

Journal articles

Brunner, C. (2019): A comparative study of Antminer+ and Decision Tree classification performances. *SEFBIS*, issue 13, pp. 15-23.

Brunner, C. (2017): Processing Intrusion Data with Machine Learning and MapReduce. *ACADEMIC AND APPLIED RESEARCH IN MILITARY AND PUBLIC MANAGEMENT SCIENCE*, issue 16, pp. 37-52.

Abstracts in conference proceedings

Brunner, C. (2019): Behatolás-detektálás Tensorflow Platformon. *16. Országos Gazdaságinformatikai Konferencia* (p. 32). Budapest: NJSZT Neumann János Számítógép-tudományi Társaság GIKOF Gazdaságinformatikai Kutatási és Oktatási Fórum.

Brunner, C. (2018): Hálózati behatolás detektálás Neurális hálózatok összevonásával. *OGIK'2018 Országos Gazdaságinformatikai Konferencia – Az előadások összefoglalói* (p. 75). Sopron, Alexander Alapítvány a Jövő Értelmiségéért.

Brunner, C. (2017): Ant Colony Algorithm in Data Mining. *XIV. Országos GazdaságInformatikai Konferencia* (p. 31). Győr: Alexander Alapítvány a Jövő Értelmiségéért.

Brunner, C. (2016): Behatolási adatok feldolgozása gépi tanulás és MapReduce segítségével. *XIII. Országos Gazdaságinformatikai Konferencia* (pp. 25-26). Dunaújváros: DUE Press.

Conference presentations

Brunner C. (2019): Behatolás-detektálás Tensorflow Platformon- Presented at: *16. Országos Gazdaságinformatikai Konferencia;* Nov 8-9; Budapest, Hungary

Brunner, C. (2018): Hálózati behatolás detektálás Neurális hálózatok összevonásával. Presented at: *OGIK'2018 Országos Gazdaságinformatikai Konferencia;* Nov 9-10; Sopron, Hungary

Brunner, C. (2017): Ant Colony Algorithm in Data Mining. Poster presented at: *XIV. Országos GazdaságInformatikai Konferencia*; Nov 10-11; Sopron, Hungary.

Brunner, C. (2016): Behatolási adatok feldolgozása gépi tanulás és MapReduce segítségével. Presented at: *XIII. Országos GazdaságInformatikai Konferncia*; Nov 11-12; Dunaújváros, Hungary.

Other works

Brunner, C. (2019): Hálózati behatolás detektálás Neurális hálózatok összevonásával. In L. Bacsárdi, G. Bencsik, & Z. Pödör, *OGIK'2018 Országos Gazdaságinformatikai Konferencia - Válogatott közlemények*. Sopron, Hungary: Alexander Alapítvány a Jövő Értelmiségéért.

In preparation

Brunner, C., Kő, A., & Fodor, S. (2020): A novel ensemble method based on Neural Networks with hyperparameter optimization for Intrusion Detection. Manuscript

6. REFERENCES

Al-Qatf, M. *et al.* (2018) "Deep learning approach combining sparse autoencoder with SVM for network intrusion detection," *IEEE Access*. IEEE, 6, pp. 52843–52856.

Anderson, J. P. (1972) *Computer security technology planning study. volume 2*.

Anderson, J. P. (1980) "Computer security threat monitoring and surveillance," *Technical Report, James P. Anderson Company*.

Batista, G. E., Prati, R. C. and Monard, M. C. (2004) "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD explorations newsletter*. ACM, 6(1), pp. 20–29.

Beek, C. *et al.* (2019) *McAfee Labs Threats Report August 2019*. Available at: https://www.mcafee.com/enterprise/en-us/threat-center/mcafee-labs/reports.html.

Bergstra, J. S. *et al.* (2011) "Algorithms for hyper-parameter optimization," in *Advances in neural information processing systems*, pp. 2546–2554.

Bergstra, J., Yamins, D. and Cox, D. D. (2013) "Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms," in *Proceedings of the 12th Python in science conference*, pp. 13–20.

Brochu, E., Cora, V. M. and De Freitas, N. (2010) "A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," *arXiv preprint arXiv:1012.2599*.

Brunner, C. (2017) "Processing Intrusion Data with Machine Learning and MapReduce," *Academic and Applied Research in Public Management Science*, 16(1), pp. 37–52.

Budzik, J. (2019) *Many Heads Are Better Than One: The Case For Ensemble Learning*. Available at: https://www.kdnuggets.com/2019/09/ensemble-learning.html (Accessed: September 29, 2019).

Chapman, P. *et al.* (2000) "CRISP-DM 1.0: Step-by-step data mining guide," *SPSS inc*, 16.

Chawla, N. V *et al.* (2002) "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Rearch*, 16, pp. 321–357.

Damiani, J. (2019) *A Voice Deepfake Was Used To Scam A CEO Out Of $243,000, Forbes*. Available at: https://www.forbes.com/sites/jessedamiani/2019/09/03/a-voice-deepfake-was-used-to-scam-a-ceo-out-of-243000/#3efedc652241 (Accessed: June 7, 2020).

Dua, S. and Du, X. (2016) *Data mining and machine learning in cybersecurity*. CRC press.

Fayyad, U., Piatetsky-Shapiro, G. and Smyth, P. (1996) "From data mining to knowledge discovery in databases," *AI magazine*, 17(3), p. 37.

Javaid, A. *et al.* (2016) "A deep learning approach for network intrusion detection system," in *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, pp. 21–26.

Lau, F. *et al.* (2000) "Distributed denial of service attacks," in *2000 IEEE international conference on systems, man and cybernetics.*, pp. 2275–2280.

Lopez-Martin, M., Carro, B. and Sanchez-Esguevillas, A. (2019) "Variational data generative model for intrusion detection," *Knowledge and Information Systems*. Springer, 60(1), pp. 569–590.

Molina-Coronado, B. *et al.* (2020) "Survey of Network Intrusion Detection Methods from the Perspective of the Knowledge Discovery in Databases Process," *arXiv preprint arXiv:2001.09697*.

Nguyen, H. M., Cooper, E. W. and Kamei, K. (2009) "Borderline over-sampling for imbalanced data classification," in *Proceedings: Fifth International Workshop on Computational Intelligence & Applications*, pp. 24–29.

Samuel, A. L. (1959) "Some Studies in Machine Learning Using the Game of Checkers," *IBM Journal of*

Research and Development, 3(3), pp. 210–229.

Scarfone, K. and Mell, P. (2007) "Guide to Intrusion Detection and Prevention Systems (IDPS) Recommendations of the National Institute of Standards and Technology," *Nist Special Publication*, 800–94, p. 127. doi: 10.6028/NIST.SP.800-94.

Smith, S. (2014) *5 Famous Botnets that held the internet hostage*. Available at: https://tqaweekly.com/episodes/season5/tqa-se5ep11.php (Accessed: June 7, 2020).

Smolyakov, V. (2017) *Ensemble Learning to Improve Machine Learning Results*. Aug. Available at: https://blog.statsbot.co/ensemble-learning-d1dcd548e936 (Accessed: March 20, 2019).

Snoek, J., Larochelle, H. and Adams, R. P. (2012) "Practical bayesian optimization of machine learning algorithms," in *Advances in neural information processing systems*, pp. 2951–2959.

Statt, N. (2019) *Thieves are now using AI deepfakes to trick companies into sending them money*, *The Verge*. Available at: https://www.theverge.com/2019/9/5/20851248/deepfakes-ai-fake-audio-phone-calls-thieves-trick-companies-stealing-money (Accessed: June 7, 2020).

Stolfo, S. J. *et al.* (2000) "Cost-based Modeling for Fraud and Intrusion Detection: Results from the JAM Project," in *Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00*. IEEE, pp. 130–144.

Tavallaee, M. *et al.* (2009) "A Detailed Analysis of the KDD CUP 99 Data Set," in *IEEE Symposium on Computational Intelligence for Security and Defense Applications - CISDA*. IEEE, pp. 1–6.

Wieringa, R. J. (2014) *Design science methodology for information systems and software engineering*. Springer.

Yang, Yanqing *et al.* (2019) "Improving the classification effectiveness of intrusion detection by using improved conditional variational autoencoder and deep neural network," *Sensors*. Multidisciplinary Digital Publishing Institute, 19(11), p. 2528.

Yin, C. *et al.* (2017) "A deep learning approach for intrusion detection using recurrent neural networks," *Ieee Access*. IEEE, 5, pp. 21954–21961.