THESIS SUMMARY:

ANOMALY DETECTION FOR NETWORK SECURITY BASED ON STREAMING DATA

Alexander Hartl
TU Wien, Institute of Telecommunications

1. INTRODUCTION

In this thesis, we explore the critical domain of Network Intrusion Detection (NID) within high-security infrastructures, emphasizing the pivotal role of data science and Machine Learning (ML) in enhancing security measures against sophisticated cyber threats. We navigate through the challenges posed by the digital transformation of critical infrastructures, such as power distribution systems, which are increasingly reliant on Information Technology (IT). The increasing usage of IT makes these infrastructures prime targets for cyber attacks, necessitating advanced security measures where Network Intrusion Detection Systems (NIDSs) play a crucial role. Given the adversarial nature of cyber security in high-security environments and the unique challenges of processing network data, we focus on the application of ML techniques to improve NID effectiveness and reliability. Building on identified gaps in existing research, this thesis is dedicated to addressing the following critical research questions:

RQ1: Which state-of-the-art anomaly detectors are best suitable for detecting network attacks in streaming data?

RQ2: How can predictions of supervised network traffic classifiers be made explainable?

RQ3: How can explainability be accomplished for unsupervised network traffic classifiers?

RQ4: Which implications does the use of extensive cryptographic techniques have for the attack surface in communication networks?

Our approach integrates a detailed examination of outlier detection in network data, focusing on both unsupervised and supervised ML methods to detect and analyze cyber threats, and considering the specific characteristics that NID yields. While NIDSs currently usually are based on supervised techniques, only unsupervised ML has the potential to detect unknown attacks and defend against sophisticated, highly targeted attacks. We prioritize explainability and interpretability in ML applications to ensure actionable insights can be derived from anomaly detection and classifier predictions. Furthermore, we delve into the complexities introduced by encryption in network security, proposing novel methods for analyzing encrypted traffic.

The contributions of this thesis are multifaceted:

- We survey existing outlier detectors for streaming data and contribute a framework providing efficient implementations to be used in various research fields.
- We identify effective combinations of outlier detection algorithms and feature vectors for network attack detection, observing the greater challenge of unsupervised methods compared to supervised ones, which are frequently used for NID.
- We develop a new outlier detector optimized for network data, addressing challenges such as performance, efficiency, interpretability, and adaptability. We emphasize network data's temporal patterns, devising novel analysis techniques, which enhance interpretability and accuracy.
- In addition to our focus on unsupervised ML interpretability, we assess the use of explainability methods in supervised IDSs, documenting their ability to reveal undesired classifier behaviors. We also adapt these methods for RNNs, which are particularly interesting in IDS contexts.

This is a summary of the doctoral dissertation [16]. Please find the full thesis under DOI 10.34726/hss.2023.98524.

- We introduce a novel method for analyzing encrypted tunnel traffic, potentially enhancing future IDS capabilities.
- We present case studies on previously unknown security challenges linked to cryptographic techniques, including a new subliminal channel in GCM encryption and innovative encrypted tunnel traffic analysis methods. These findings open new research avenues and urge cautious encryption application.

Our work underscores the necessity of integrating advanced data science techniques into the cyber security strategy of critical infrastructures. By pushing the boundaries of current understanding and application in the field of NID, this research sets a foundation for future exploration and development in securing high-security environments from sophisticated cyber attacks.

2. APPLYING STREAM OUTLIER DETECTION TO NETWORK DATA

Many research publications name cyber attack detection as paramount example of anomaly detection. However, to date the intuitive fact that network attacks present themselves as outliers has not been established. Exploring the potential of unsupervised methods for attack detection within NID requires a dual focus: examining streaming outlier detection techniques suitable for the dynamic nature of network traffic and determining the extent to which network attacks constitute outlying data samples detectable by these methods.

Our research involves consolidating existing outlier detection approaches, enhancing implementation efficiency, evaluating different feature vectors, and assessing methods' capabilities to handle concept drift and high-rate streaming data. By surveying state-of-the-art techniques, developing a framework for their efficient application (dSalmon), and conducting experimental evaluations, this work aims to address the outlined research question RQ1, focusing on the applicability and performance of unsupervised IDS methodologies in detecting network attacks through outlier analysis.

2.1. *A Comparative Study on Stream Outlier Detection Techniques*

In various applications, such as fraud detection, network security, and medical monitoring, data streams with independent multivariate samples arrive continuously at unpredictable rates. Detecting outliers—or anomalous samples—within these streams is crucial for identifying potential threats or anomalies. Despite the plethora of methods proposed for this task, there lacks a comprehensive comparison of these techniques, especially in handling the dynamic nature of data streams.

We present a systematic examination of outlier detection methods tailored to streaming data, focusing on their core mechanisms and adaptability to concept drift—a phenomenon where the statistical properties of the target data gradually change over time. We categorize the techniques based on their approaches to managing concept drift and identifying outliers. In particular, we identify the following approaches for managing concept drift:

- **Sliding Windows (SWs)**: A commonly used method, SWs consider only the most recent data points, either by maintaining a constant number of recent samples or by ensuring a fixed memory span. SWs are straightforward but may struggle with non-uniform data arrival rates and density fluctuations.
- **Reference Windows (RWs)**: RWs involve occasional model retraining based on the most recent samples. They allow for computationally intensive models due to less frequent updates but may lag in adapting to new data.
- **Exponential Moving Averaging**: This method updates models using Exponentially Weighted Moving Averages (EWMA), enabling swift adaptation without the need to store past samples.

To construct a stream outlier detector, one of these approaches is combined with one of the following approaches for identifying outliers, which provides the fundamental definition of outlierness:

- **Distance-based:** Utilizes the proximity of data points in feature space to score outliers. This approach is straightforward and interpretable but computationally intensive for dynamic data streams.
- **Histogram-based:** Builds probabilistic models from data distributions, with outliers being those in low-probability areas. It offers simplicity but may miss complex patterns.

	SW-DBOR	SW-KNN	SW-LOF	Loda [28]	RS-Hash [29]	RRCF [15]	HS-Trees [32]	xStream [25]	SDOstream [18]
Windowing mechanism	SW	SW	SW	SW	SW	SW	RW	RW	EWMA
Constant space complexity	×	×	×	×	×	×	✓	✓	✓
Constant time complexity	×	×	×	×	✓	×	✓	✓	✓
Ensemble-based	×	×	×	✓	✓	✓	✓	✓	×
Parallelizable in dSalmon	×	×	×	✓	✓	✓	✓	✓	×

FIG. 1.— Characteristics of streaming outlier detectors.

- **Binning-based:** Generalizes histograms by dividing the feature space into bins, with outliers determined by sparse bins. It is adaptable but loses the probabilistic interpretation.
- **Density-based:** Considers local data density for outlier detection, potentially offering detailed insights but at significant computational costs.
- **Isolation-based:** Identifies outliers by their ease of isolation from the rest of the data, a concept used in methods like the Robust Random Cut Forest [15].

In evaluating these methods, we prioritize their adaptability to concept drift, computational efficiency, and interpretability. Distance-based methods stand out for their interpretability and straightforwardness, although they may not always be the most computationally efficient. Histogram and binning-based methods offer a balance between performance and simplicity but may sacrifice detailed understanding of the data. Advanced techniques like tree structures and isolation-based methods provide innovative solutions to adaptability and computational challenges, albeit at the expense of straightforward interpretability. We show in Fig. 1 an overview of streaming outlier detection methods, highlighting characteristics influencing computational demands. We incorporate dSalmon and SDOstream in this figure, which we describe in the upcoming sections.

With this work, we underscore the diversity and complexity of outlier detection in streaming data, highlighting the trade-offs between adaptability, computational demands, and interpretability. Future research should focus on hybrid approaches that leverage the strengths of these methods while mitigating their weaknesses, especially in rapidly evolving data streams.

2.2. *Efficient Anomaly Detection in Data Streams with dSalmon*

In the rapidly evolving digital landscape, the ability to efficiently process and analyze data streams is crucial, especially for tasks like anomaly detection. The introduction of dSalmon marks a significant advancement in this context. As a Python-compatible framework, dSalmon addresses the limitations of existing solutions like PySAD [34], including slow processing speeds and scalability issues, by leveraging C++ for its core operations. In particular for researchers, slow processing times significantly hinder productivity, particularly because algorithm parameter tuning often requires executing a large number of runs across diverse datasets. This challenge can render some methods practically unusable due to the prohibitive nature of their processing inefficiencies.

dSalmon's design choice results in marked improvements in execution times and memory efficiency. It distinguishes itself from traditional methods and frameworks by its focus on high-speed processing and ease of use, making it uniquely suited for the analysis of evolving data streams, as opposed to static datasets or systems with limited capabilities for stream analysis.

dSalmon stands out due to its architectural design and interface, tailored for ease of use and high performance. To this end, we have made design decisions as follows:

C++ Core for Speed: By implementing the core algorithms in C++, dSalmon achieves significant performance gains, processing data up to three orders of magnitude faster than PySAD.

Python Interface for Usability: Despite its C++ backbone, dSalmon provides a Pythonic interface, ensuring ease of use for data scientists familiar with Python.

Efficient Handling of Data Streams: dSalmon's design allows for efficient processing of streaming data, overcoming the challenges posed by the need for continuous model adaptation.

Support for Ensemble Learning and Parallel Processing: The framework supports efficient ensemble learning and takes full advantage of modern multi-core processors to accelerate computations.

Minimal Dependencies: dSalmon requires only NumPy for installation, simplifying setup and reducing potential compatibility issues.

A comprehensive evaluation of dSalmon against PySAD reveals dSalmon's superior performance in terms of execution time, memory usage, and CPU energy consumption. We have performed benchmarks on three popular datasets, and demonstrated its capability to process millions of data samples swiftly and accurately. Our experiments show that dSalmon is able to process datasets between one and three orders of magnitude faster than established Python implementations. We documented our architectural decisions and benchmarking results in [20], and performed a rigorous performance comparison of established outlier detectors [24]. We released dSalmon in our GitHub repository[1] as open source software under the LGPL license.

2.3. *Evaluating Outlier Detection for NID*

Network security analytics focuses on attack detection, anomaly detection, and traffic classification. These areas often intersect, creating challenges in differentiating between normal network behavior and potential threats. While Distributed Denial-of-Service (DDoS) attacks may present as anomalies in time-series analysis, they might not appear so in a spatial context due to their prevalence. This raises the question: Are network attacks genuinely outliers or just common occurrences that blend into regular traffic patterns?

The distinction between "anomaly" and "outlier" is nuanced and impacts the applicability of unsupervised methods for attack detection. Network attacks, designed to evade detection, may not always manifest as clear outliers. This ambiguity necessitates a precise definition of "anomaly" to avoid high false-positive rates, which hinder the adoption of ML techniques in NID. Given that IDS datasets often feature an unrealistically high ratio of attack to normal traffic, this discrepancy can significantly affect the performance of unsupervised detection methods.

We explore the outlierness of network attacks through experiments with five space representations and unsupervised outlier detection algorithms. We aim to identify which feature representations best pinpoint attacks as outliers and whether detected outlierness is a reliable indicator for real-world attack detection. The experiments are conducted using the CIC-IDS-2017 [30] dataset, focusing on static data analysis due to dataset limitations. Our findings can be summarized as follows:

Outlierness of Network Attacks: Attacks tend to form clusters distant from normal traffic, suggesting they are global, clustered outliers. Algorithms like Histogram-based Outlier Score (HBOS) [13] and Sparse Data Observers (SDO) [23] perform best due to their ability to interpret space globally.

Feature Representation: The AGM [21] vector format shows the most significant differences between attack and non-attack instances, while formats like CAIA [2] and Consensus [11] offer complementary views. A new vector, OptOut, is proposed to maximize the separation between attacks and non-attacks, combining the strengths of existing feature vectors.

Real-world Applicability: Although the OptOut vector enhances the distinction between normal and malicious traffic, solely relying on unsupervised detection algorithms may not suffice for real-world applications due to the potential for high false-positive rates.

Hence, network attacks exhibit higher outlierness compared to normal traffic when analyzed with suitable feature vectors and algorithms, particularly HBOS and SDO. However, the overlap between the outlierness of attacks and legitimate traffic necessitates careful consideration. The proposed OptOut vector improves performance, but unsupervised methods alone cannot guarantee effective real-world attack detection. Combining unsupervised approaches with supervised methods and leveraging pre-existing knowledge could lead to more accurate and reliable NIDS. We documented these findings in [22].

[1] https://github.com/CN-TU/dSalmon

TABLE 1
PERFORMANCE COMPARISON WITH DIFFERENT OUTLIER DETECTION ALGORITHMS.

	SWAN-SF [3]			**KDD Cup'99** [1]		
	AAP	AP@n	AUC	AAP	AP@n	AUC
SW-KNN	0.69	**0.56**	**0.91**	0.07	0.15	0.72
SW-LOF	0.15	0.12	0.58	-0.00	-0.00	0.67
Loda [28]	0.72	0.54	**0.91**	0.10	0.13	0.92
RS-Hash [29]	**0.73**	0.55	**0.91**	0.13	0.15	0.95
RRCT [15]	0.23	0.19	0.69	0.07	0.05	0.85
Our method	**0.73**	0.55	**0.91**	**0.33**	**0.54**	**0.97**

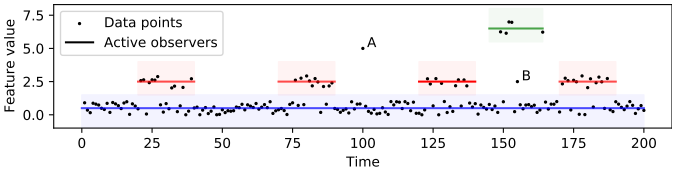

FIG. 2.— Illustration of out-of-phase outliers. While A is a spatial outlier, B is an out-of-phase outlier.

3. NOVEL ANALYSIS METHODS

We introduce SDOstream as a novel outlier detection algorithm tailored for streaming data, addressing the challenge of monitoring network data in real-time in an interpretable way (RQ3). It is built upon the understanding gained from addressing RQ1, offering a solution that avoids the drawbacks of traditional methods like kNN and LOF, which, when used with an SW, require recalling all previous data up to a certain point for processing new data points. SDOstream operates efficiently in linear time with respect to the number of processed data points, eliminating the need for a SW and hence, overcoming the limitations of memory length dictated by algorithm constraints. Instead, it allows memory length to be determined based on application needs.

The algorithm is designed to adapt dynamically to streaming data by utilizing a set of sampled data points named *observers* within the model. These observers are evaluated and updated as new data arrives, using parameters such as the fading parameter (f) to balance model adjustment speed against noise stability. This approach avoids constant retraining, addressing the challenge of concept drift in streaming data. SDOstream's design ensures that outlierness is defined in an intuitive and interpretable manner, based on spatial distances and the locations of representative points.

In terms of technical implementation, SDOstream maintains a fixed-size model, leading to space and time complexities of $O(k)$ and $O(n \log k)$, respectively, where n denotes the number of processed samples and k denotes the model size. In a nutshell, the algorithm can be summarized as

1. Find the sets of nearest observers and nearest *active* observers, i.e. observers in regions with sufficient density.
2. Update model statistics to keep track of point densities around observers. Sample new points regularly.
3. Use the nearest active observers set for scoring outlierness based on the median distance to observers.

SDOstream can leverage data structures like M-Trees [10] for efficient nearest neighbor search, enabling the algorithm to scale to large data streams. Performance evaluations using real and synthetic datasets demonstrate SDOstream's effectiveness in detecting outliers, showcasing its fast, versatile capabilities and stability across various scenarios. It shows comparable or superior performance to traditional outlier detection methods, underscoring its potential in handling large volumes of streaming data with evolving characteristics.

SDOstream exemplifies a significant advancement in outlier detection for streaming data, providing a practical, efficient solution that addresses the specific needs of network data analysis. Its adaptive, model-based approach offers a scalable, interpretable strategy for real-time anomaly detection, making it a valuable tool for various applications requiring continuous data monitoring and analysis.

3.1. *Mining Periodic Patterns*

We introduce an extension to SDOstream for detecting anomalies in streaming data by incorporating temporal patterns to identify out-of-phase outliers—data points that deviate from established temporal patterns. We show an illustration in Fig. 2. Fig. 2 depicts three clusters of different periodicities. These clusters are captured by different observers, which temporarily reappear based on a learned pattern. B occurs outside of the usual pattern and therefore is an out-of-phase outlier.

This novel approach not only enhances anomaly detection but also provides insights for descriptive analysis and knowledge extraction from temporal patterns in data streams and, hence, is ideally suited for network data processing

in high-security infrastructures. The extended algorithm, building upon the SDOstream framework, integrates Fourier transforms, blending into an EWMA to model temporal patterns without increasing asymptotic space or time complexity. It captures periodicities in data streams, making it possible to dynamically adjust the model to the stream's temporal dynamics and detect anomalies based on deviations from these patterns.

The unique advantage of this work lies in its capacity to capture and identify temporal patterns, setting it apart from conventional outlier detection approaches. Key contributions compared to existing outlier detectors retain the benefits SDOstream yields, including scalability, with the algorithm efficiently processing high-rate data streams without increasing complexity with memory length; interpretability, providing clear rationale for anomaly detection through spatial and temporal relationships; and accuracy, demonstrating improved detection of out-of-phase outliers.

The algorithm's core involves updating a fixed-size model with temporal functions representing the periodic observation of data points, allowing for dynamic adjustment to the stream's temporal patterns. This approach ensures scalability and interpretability, crucial for applications requiring understanding of anomaly detection decisions. Experimental evaluations show the algorithm's effectiveness in capturing temporal patterns and identifying out-of-phase outliers, demonstrating superior performance compared to existing methods in certain contexts. In particular, Table 1 shows adjusted average precision (AAP), adjusted P@n (AP@n) and ROC-AUC for two datasets, demonstrating that in particular KDD Cup'99 [1] exhibits patterns that can effectively be used to improve outlier detection. Evaluation metrics are adopted from [7].

Furthermore, the method's application to real-world scenarios, such as network traffic analysis, showcases its ability to uncover meaningful temporal patterns and anomalies, underscoring its practical utility for knowledge discovery in streaming data. As example, we show in Fig. 3 learned temporal information for an observer when processing data we have captured in an e-charging infrastructure. Information as depicted in this figure can be retrieved from the model at any point in time, i.e. without having to invest any additional time monitoring the data stream in the indicated region in feature space.

The algorithm's design, emphasizing scalability, interpretability, and accuracy, positions it as an advancement in stream data analysis, particularly in environments where understanding temporal dynamics is key to anomaly detection and knowledge discovery. In particular for network traffic analysis, the unique combination of properties makes this work ideally suited for practical applications. While we introduced SDOstream in [18], we consider low-density models a promising foundation for further data mining problems and we still explore new ways to leverage their full potential.

3.2. *Encrypted Flow Separation*

To approach RQ4, we delve into the challenge of analyzing encrypted network traffic by separating packets belonging to different flows without being able to decrypt the content. To the best of our knowledge, this endeavor has not been addressed in related work, signifying a substantial advancement of the state of the art.

Our approach involves two main steps: developing a potent anomaly detector to identify anomalous packets within a flow and determining the separation of flows that minimizes total anomalousness. Our methodology leverages packet features such as arrival time, length, and direction, with packets represented as feature vectors. For anomaly detection, a deep neural network (NN) utilizing LSTM units is employed to predict the probability distribution of the next packet's features, providing the ability to identify unusual sequences of packet features. The second step involves using maximum likelihood estimation to find the flow separation that minimizes total anomalousness, based on an algorithm inspired by Viterbi [33] but tailored for continuous state spaces. We also explore various modifications and extensions of the method, including omitting certain features, predicting features of the next two packets, and utilizing a backward model for improved separation.

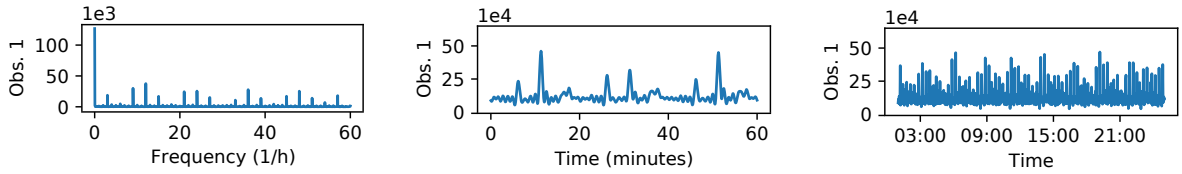

FIG. 3.— Learned magnitude spectrum (left), one-hour temporal plot (center) and 24-hour temporal plot (right) for an exemplary observer when processing network data captured in an e-charging infrastructure.

TABLE 2
FLOW SEPARATION RESULTS FOR MAWI [9] (LEFT), CIC-IDS-2017 [30] (CENTER) AND UNSW-NB15 [27] (RIGHT).

	[9], pkt. avg.		[9], seq. avg.		[30], pkt. avg.		[30], seq. avg.		[27], pkt. avg.		[27], seq. avg.	
Flows	**Acc.**	**TrAcc.**	**Acc.**	**TrAcc.**	**Acc.**	**TrAcc.**	**Acc.**	**TrAcc.**	**Acc.**	**TrAcc.**	**Acc.**	**TrAcc.**
2	0.983	0.977	0.990	0.986	0.996	0.996	0.997	0.997	0.998	0.998	0.998	0.998
5	0.932	0.926	0.945	0.944	0.981	0.984	0.984	0.985	0.991	0.989	0.991	0.990

Experiments on both synthetic and real-world datasets demonstrate the method's effectiveness in separating encrypted traffic into their original flows. Real-world data results are shown in Table 2. Here, accuracy measures whether packets in a flow are matched correctly, while transition accuracy (TrAcc) measures whether flow transitions of subsequent packets are detected correctly. Since flows have different lengths, averaging can be performed on a per-packet and per-sequence basis, resulting in different performance readings. Despite the common understanding that separating flows in encrypted traffic is a hard problem, we find that impressively high performance results can be obtained, as indicated by Table 2. Hence, despite the encrypted nature of the traffic, distinct patterns within individual flows can be exploited for successful separation, challenging the perceived security and privacy benefits of encryption techniques used in modern communication networks. This revelation underscores the need for careful engineering beyond simple encryption to ensure strong security and privacy properties in network protocols. We disseminated these findings in [19].

4. EXPLAINABILITY FOR IDSS USING SUPERVISED ML

Explainability in ML-based Intrusion Detection Systems (IDSs) is crucial, particularly in high-security environments where understanding ML decisions can influence adoption. Supervised learning, while effective in attack detection given labeled data, often lacks in providing clear explanations for its predictions. This gap has led to the development of innovative methods aimed at interpreting these decisions, especially pertinent in the context of IDSs where explanations are vital for trust and regulatory compliance.

Recent advancements in explainability for IDSs leverage both established and novel approaches. For instance, examining statistical features of network traffic through supervised ML models reveals patterns and anomalies in data, yet the complexity of modern NN architectures complicates interpretability. To address this in the course of RQ2, we propose new explainability techniques designed to improve understanding of NN decisions, even when classifiers handle complex data like full packet features.

An interesting area of exploration is the detection and interpretation of adversarial examples and poisoning attacks. By applying adversarial sample generation techniques to Recurrent Neural Networks (RNNs) used for NID, we assess the robustness of IDS classifiers against such threats. Additionally, we explore the utility of explainability plots, such as Partial Dependence Plots (PDPs) [12] and Accumulated Local Effects (ALE) [4] plots, for identifying vulnerabilities in ML models, including hidden backdoors that could bypass attack detection mechanisms.

Moreover, our research extends explainability methods to sequential data, crucial for analyzing encrypted traffic at the packet level. In particular, we introduce the *sequential* PDP for feature i

$$\text{seqPDP}_{c,i}(t,w) = \mathbb{E}_{X|C}\Big(f\left(h_{t-1}(X), X_1^t, \ldots, X_{i-1}^t, w, X_{i+1}^t, \ldots X_n^t\right)|c\Big). \tag{1}$$

with input data X^t, hidden states $h_t(X)$ and class c, allowing to visualize predictions made by RNNs. By evaluating feature importance and sensitivity through various methods, including NN weights, input perturbation, and feature dropout, we identify key features that significantly impact classification decisions. These insights enable a more nuanced understanding of model behavior, highlighting the importance of certain features over others while also uncovering potential avenues for adversarial manipulation.

We show an example in Fig. 4. Here, we have created adversarial samples based on the Carlini-Wagner (CW) algorithm [8], and compared manipulations applied by CW to regions that indicate a change of classifier output based on a sequential PDP. In the depicted example, adversarial manipulations agree largely with insights obtained from sequential PDPs. While this shows the potential that explainability techniques yield for robustness in NID domains, we also note that we did not observe a strong agreement for many other adversarial samples, reinforcing the complexity of understanding ML models with high-dimensional inputs.

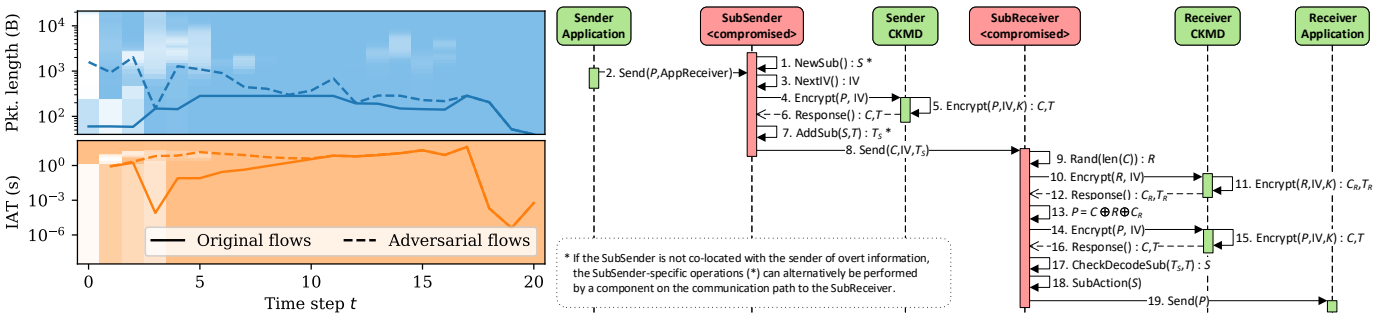

FIG. 4.— Exemplary sequential PD plot (left) and our procedure in exploiting the AES-GCM subliminal channel (right).

The development of an RNN-based IDS exemplifies the challenges and opportunities in applying ML to security. Our experiments with sequential PDPs additionally reveal the potential for early detection of attacks within a flow's lifetime, emphasizing the classifier's ability to make accurate predictions based on initial packets. However, the susceptibility to adversarial examples highlights the need for careful model evaluation and the potential preference for more interpretable classifiers in certain security contexts.

Enhancing explainability in ML-based IDSs is paramount for their effective and trustworthy deployment in security-sensitive environments. By combining traditional and novel explainability techniques, our research advances the understanding of ML decisions in IDSs, paving the way for more transparent, reliable, and robust intrusion detection mechanisms. This exploration not only contributes to the academic discourse on ML explainability but also offers practical guidelines for developing and evaluating secure and interpretable IDSs. Our publications [6, 17] document these findings, applying explainability techniques to counter the threats of poisoning attacks and adversarial ML.

5. ATTACKS ON HIGH-SECURITY INFRASTRUCTURES

The utility of encryption in safeguarding digital communication is undeniable, yet its application can inadvertently introduce vulnerabilities. To complete our discourse on RQ4, we delve into a specific novel vulnerability associated with AES-GCM [26]—a popular mode of operation for AES encryption—when deployed in critical infrastructures alongside specific capsulated security devices, we refer to as Cryptographic Key Management Devices (CKMDs). The vulnerability facilitates subliminal communication, undermining the integrity of high-security networks. Subliminal channels in cryptographic systems, first conceptualized by Simmons [31], exploit the very mechanisms designed for security to enable covert communication. We focus on AES-GCM's Message Authentication Code (MAC) to demonstrate how these channels can be established, even in environments designed to mitigate cryptographic attacks.

AES-GCM is favored for its efficiency and security. However, we reveal that its MAC, under certain network architectures designed to offload cryptographic operations to CKMDs, can be exploited for subliminal communication. This exploitation is made possible due to architectural choices that, while aiming to enhance security, inadvertently enable these covert channels. By manipulating the authentication tag generated in the AES-GCM encryption process, data can be embedded within the tag, escaping detection by traditional security mechanisms. This process involves intricate manipulations of cryptographic operations, which we show in Fig. 4. As an illustrative example, we analyzed an IIoT infrastructure, which is reliant on AES-GCM for data security, but might become vulnerable to subliminal communication. Despite the presence of CKMDs intended to centralize and secure cryptographic operations, the architecture facilitates a bypass that can be exploited for covert data transmission.

Addressing this vulnerability requires a multifaceted approach. Recommendations include adopting alternative encryption modes like GCM-SIV [14], which is less susceptible to such exploitation, and revising the operational protocol of CKMDs to prevent manipulation of the IV or authentication tags. Additionally, employing separate keys or IVs for different communication directions and ensuring that CKMDs enforce strict controls on cryptographic operations are vital steps in mitigating the risks.

Hence, while encryption plays a crucial role in securing digital communications, its implementation must be carefully considered within the architecture of high-security networks. The discovery of subliminal channels in AES-GCM underscores the complexity of cyber security, highlighting the need for continuous evaluation and adaptation of security practices to address evolving threats. We disseminated these findings in [5].

6. DISCUSSION

This thesis explores the design of ML-based NIDSs for securing critical infrastructures, emphasizing the importance of both unsupervised and supervised learning methods. Unsupervised learning, particularly through streaming outlier detection algorithms, reveals limitations in detection performance and interpretability when applied to network traffic. Our dSalmon framework attempts to address these limitations, alongside the introduction of specialized feature vectors like OptOut for optimizing detection performance. However, the innate potential of unsupervised learning lies in its capability to uncover unknown attacks, making interpretability a critical factor for manual data investigation.

Conversely, supervised learning demonstrates significant promise in detection performance, with this research endeavoring to enhance explainability in these systems. Techniques providing inherent interpretability are preferred over auxiliary methods, PDPs or ALEs, which only offer limited insights into the decision-making processes of ML models.

6.1. *Challenges and Future Research Directions*

A major problem in ML-based IDS research is the scarcity of suitable, comprehensive datasets, which is exacerbated by data protection concerns and the intensive resources required for data curation and labeling. This thesis primarily utilized the CIC-IDS-2017 and UNSW-NB15 datasets, despite their known limitations. The lack of comparable metrics across studies further complicates the evaluation of IDS performance, a challenge this work attempts to address by employing a variety of metrics.

Future research should focus on developing more effective anomaly detection methods and generating high-quality datasets, possibly through frameworks that allow for dataset evolution over time. The exploration of encrypted traffic flow identification presents a novel avenue for IDS application, highlighting the need for further investigation into NN architectures and traffic separation techniques.

6.2. *Recommendations for IDS Construction*

Our findings suggest that a hybrid approach, combining unsupervised and supervised ML methods, may offer the most effective solution for IDS deployment in network intrusion detection. Such systems should prioritize interpretability to aid human analysts in decision-making processes. Additionally, incorporating temporal information about traffic patterns could significantly enhance the understanding of network activities.

The identification of traffic flows within encrypted tunnels emerges as a promising strategy, yet it also underscores the necessity for improved traffic obfuscation techniques to counteract potential attackers. This thesis advocates for a comprehensive security strategy that extends beyond IDS deployment, emphasizing the importance of preventative measures to block malware infiltration into networks.

In summary, while ML-based IDSs hold considerable potential for improving network security, their development and deployment must carefully consider the balance between detection capabilities, interpretability, and the challenges posed by the current state of data and algorithmic limitations.

7. CONCLUSIONS

With this thesis, we explore the crucial role of network security research within high-security infrastructures, emphasizing not only the need for high accuracy in attack detection but also the importance of explainability and the capability to identify sophisticated, previously unknown attacks. We delved into several key areas, including the significance of outlier detection in streaming data through unsupervised methods for uncovering unknown attacks, and the development of algorithms that contribute to the current research landscape. Our work in explainability and interpretability across supervised and unsupervised domains, coupled with the creation of a novel algorithm for unsupervised anomaly detection suited for NID challenges, underscores our contributions. Despite these advancements, our work highlights ongoing challenges in IDS evaluation and in unsupervised attack detection. We introduced several novel ideas such as the GCM subliminal channel attack scenario and approaches for encrypted traffic analysis and temporal pattern handling, opening new research directions. While unsupervised methods currently show limited detection accuracy, our anomaly detector is well-balanced between interpretability and efficiency. In general, our investigations suggest that simpler methods might be more effective in high-security contexts than complex models. While efforts towards explainability of supervised ML are valuable, they do not fully resolve the complexities of decision-making

in advanced classifiers. Our work emphasizes the ongoing need for rigorous security practices and the possibility of undiscovered risks even in well-secured IT infrastructures.

REFERENCES

[1]Kdd cup 1999 data. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, 1999. Accessed: 2023-05-07.

[2]Riyad Alshammari and A Nur Zincir-Heywood. A preliminary performance comparison of two feature sets for encrypted traffic classification. In *Proc. of the Int. Workshop on Computational Intelligence in Security for Information Systems CISIS'08*, pages 203–210. Springer, 2009.

[3]Rafal A. Angryk, Petrus C. Martens, Berkay Aydin, Dustin Kempton, Sushant S. Mahajan, Sunitha Basodi, Azim Ahmadzadeh, Xumin Cai, Soukaina Filali Boubrahimi, Shah Muhammad Hamdi, Michael A. Schuh, and Manolis K. Georgoulis. Multivariate time series dataset for space weather data analytics. *Scientific Data*, 7(227), 2020.

[4]Daniel W. Apley and Jingyu Zhu. Visualizing the Effects of Predictor Variables in Black Box Supervised Learning Models. *arXiv:1612.08468 [stat]*, December 2016. arXiv: 1612.08468.

[5]Marcel Armour and Bertram Poettering. Subverting decryption in AEAD. Technical Report 987, 2019. URL http://eprint.iacr.org/2019/987.

[6]Maximilian Bachl, Alexander Hartl, Joachim Fabini, and Tanja Zseby. Walling Up Backdoors in Intrusion Detection Systems. In *Big-DAMA '19*, pages 8–13, Orlando, FL, USA, 2019. ACM.

[7]G. O. Campos, A. Zimek, et al. On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data Mining and Knowledge Discovery*, 30 (4):891–927, 2016. ISSN 1573-756X. doi:10.1007/s10618-015-0444-8.

[8]Nicholas Carlini and David Wagner. Towards Evaluating the Robustness of Neural Networks. In *S&P*, pages 39–57. IEEE, May 2017.

[9]Kenjiro Cho, Koushirou Mitsuya, and Akira Kato. Traffic data repository at the WIDE project. In *2000 USENIX Annual Technical Conference (USENIX ATC 00)*, 2000.

[10]P. Ciaccia, M. Patella, et al. M-tree: An efficient access method for similarity search in metric spaces. In *Proc. of the 23rd VLDB conference*, pages 426–435, 1997.

[11]Daniel C. Ferreira, Félix Iglesias Vázquez, Gernot Vormayr, Maximilian Bachl, and Tanja Zseby. A meta-analysis approach for feature selection in network traffic research. In *Proc. of the Reproducibility Workshop*, Reproducibility '17, pages 17–20. ACM, 2017. ISBN 978-1-4503-5060-0.

[12]Jerome H. Friedman. Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics*, 29(5): 1189–1232, 2001.

[13]Markus Goldstein and Andreas Dengel. Histogram-based outlier score (HBOS): A fast unsupervised anomaly detection algorithm. *KI 2012: Advances in artificial intelligence: 35th Annual German Conference on AI*, pages 59–63, 2012.

[14]Shay Gueron and Yehuda Lindell. GCM-SIV: Full nonce misuse-resistant authenticated encryption at under one cycle per byte. Technical Report 102, 2015. URL http://eprint.iacr.org/2015/102.

[15]S. Guha, N. Mishra, et al. Robust random cut forest based anomaly detection on streams. In *Proc. of The 33rd Int. Conf. on Machine Learning*, volume 48 of *Proc. of Machine Learning Research*, pages 2712–2721, New York, USA, 2016. PMLR.

[16]Alexander Hartl. *Anomaly Detection for Network Security based on Streaming Data*. PhD thesis, Technische Universität Wien, 2023.

[17]Alexander Hartl, Maximilian Bachl, Joachim Fabini, and Tanja Zseby. Explainability and adversarial robustness for RNNs. In *2020 IEEE Sixth International Conference on Big Data Computing Service and Applications (BigDataService)*, pages 148–156, New York, NY, USA, 2020. IEEE.

[18]Alexander Hartl, Félix Iglesias, and Tanja Zseby. SDOstream: Low-density models for streaming outlier detection. In *ESANN 2020 proceedings*, pages 661–666, 2020.

[19]Alexander Hartl, Joachim Fabini, and Tanja Zseby. Separating flows in encrypted tunnel traffic. In *21st IEEE International Conference on Machine Learning and Applications*, pages 609–616. IEEE, 2022.

[20]Alexander Hartl, Félix Iglesias, and Tanja Zseby. dSalmon: High-Speed Anomaly Detection for Evolving Multivariate Data Streams. In *16th EAI International Conference on Performance Evaluation Methodologies and Tools*. ACM, 2023.

[21]Félix Iglesias and Tanja Zseby. Pattern discovery in internet background radiation. *IEEE Transactions on Big Data*, 5(4): 467–480, 2017.

[22]Félix Iglesias, Alexander Hartl, Tanja Zseby, and Arthur Zimek. Are network attacks outliers? a study of space representations and unsupervised algorithms. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 159–175. Springer, 2019.

[23]Félix Iglesias Vázquez, Tanja Zseby, and Arthur Zimek. Outlier detection based on low density models. In *2018 IEEE International Conference on Data Mining Workshops, ICDM Workshops, Singapore, Singapore, November 17-20, 2018*, pages 970–979, 2018.

[24]Félix Iglesias Vázquez, Alexander Hartl, Tanja Zseby, and Arthur Zimek. Anomaly detection in streaming data: A comparison and evaluation study. *Expert Systems with Applications*, 233, 2023.

[25]E. Manzoor, H. Lamba, et al. xStream: Outlier detection in feature-evolving data streams. In *24th ACM SIGKDD Int. Conf. on Know. Discovery and Data Mining*, 2018.

[26]David McGrew and John Viega. The galois/counter mode of operation (gcm). *Submission to NIST Modes of Operation Process*, page 44, 2004.

[27]Nour Moustafa and Jill Slay. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In *2015 Military Communications and Information Systems Conference (MilCIS)*, pages 1–6, November 2015.

[28]Tomáš Pevný. Loda: Lightweight on-line detector of anomalies. *Machine Learning*, 102(2):275–304, February 2016.

[29]Saket Sathe and Charu C. Aggarwal. Subspace outlier detection in linear time with randomized hashing. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 459–468, 2016.

[30]Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In *ICISSP*, pages 108–116, Funchal, Madeira, Portugal, 2018. SCITEPRESS.

[31]Gustavus J. Simmons. The prisoners' problem and the subliminal channel. In *Advances in Cryptology – CRYPTO'83*, pages 51–67. Springer, Boston, MA, January 1984.

[32]Swee Chuan Tan, Kai Ming Ting, and Tony Fei Liu. Fast anomaly detection for streaming data. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[33]Andrew J Viterbi. A personal history of the viterbi algorithm. *IEEE Signal Processing Magazine*, 23(4):120–142, 2006.

[34]Selim F Yilmaz and Suleyman S Kozat. Pysad: A streaming anomaly detection framework in python. *arXiv preprint arXiv:2009.02572*, 2020.