Advancing Network Survivability and Reliability: Integrating XAI-Enhanced Autoencoders and LDA for Effective Detection of Unknown Attacks

Fatemeh Stodt, Fabrice Theoleyre, Christoph Reich

Advancing Network Survivability and Reliability: Integrating XAI-Enhanced Autoencoders and LDA for Effective Detection of Unknown Attacks

Fatemeh Stodt
IDACUS
Furtwangen University
Furtwangen, Germany
0000-0003-0863-0907

Fabrice Theoleyre
ICube
CNRS / University of Strasbourg
Strasbourg, France
0000-0002-7903-3520

Christoph Reich
IDACUS
Furtwangen University
Furtwangen, Germany
0000-0001-9831-2181

***Abstract*—This study presents a novel approach for fortifying network security systems, crucial for ensuring network reliability and survivability against evolving cyber threats. Our approach integrates Explainable Artificial Intelligence (XAI) with an ensemble of autoencoders and Linear Discriminant Analysis (LDA) to create a robust framework for detecting both known and elusive zero-day attacks. We refer to this integrated method as AE-LDA. Our method stands out in its ability to effectively detect both known and previously unidentified network intrusions. By employing XAI for feature selection, we ensure improved interpretability and precision in identifying key patterns indicative of network anomalies. The autoencoder ensemble, trained on benign data, is adept at recognising a broad spectrum of network behaviours, thereby significantly enhancing the detection of zero-day attacks. Simultaneously, LDA aids in the identification of known threats, ensuring a comprehensive coverage of potential network vulnerabilities. This hybrid model demonstrates superior performance in anomaly detection accuracy and complexity management. Our results highlight a substantial advancement in network intrusion detection capabilities, showcasing an effective strategy for bolstering network reliability and resilience against a diverse range of cyber threats.**

***Index Terms*—Network Anomaly Detection, Autoencoders, Unsupervised Learning, Network Reliability, Cybersecurity**

I. Introduction

In an era where the digital domain has become an integral part of our daily lives, the reliability and survivability of network systems stand as crucial elements in ensuring the continuity and security of our digital interactions. The uninterrupted operation and safeguarding of these systems against various cyber threats form the backbone of a resilient network environment [1]. Central to this resilience is the ability to detect anomalies, which are departures from expected network behaviour, ranging from benign errors to malicious cyberattacks and data breaches [2].

The early identification of such anomalies is vital in preventing attacks and maintaining the integrity of network operations. Traditional detection methods, which focus on known patterns and signatures of attacks [3], often sourced from honeypots [4], are increasingly insufficient in the face of sophisticated and unprecedented threats, notably zero-day attacks. These attacks, lacking any prior signature, pose a significant challenge to the reliability and survivability of network systems.

In response to these challenges, recent advancements have turned towards Machine and Deep Learning techniques [5], such as Bidirectional Long Short Term Memory (LSTM) [6], to capture expected behaviours in network traffic. However, defining and interpreting these behaviours on a per-flow basis introduces complexities in accurately identifying intricate attack patterns. Moreover, prediction models exist for aggregate traffic [7], but it is much more difficult to characterize each traffic flow to detect anomalies.

In this context, autoencoders, a Deep Learning technique, emerge as a promising solution for anomaly detection in network security. They are trained on standard network traffic to develop an understanding of 'normal' network behaviour, enabling them to effectively identify deviations. Inspired by image processing, Chen *et al.* [8] implement convolutional/deconvolutional layers before applying the autoencoder. However, network information is not as spatial as the colocated pixels of an image.

In this paper, we present a novel approach using an autoencoder enhanced with Linear Discriminant Analysis, hereafter referred to as AE-LDA, The contributions of this paper are as follows:

1) Advanced Anomaly Detection: Our model, built on existing machine learning techniques, accurately detects and classifies network anomalies
2) Versatile and Robust Performance: Our model excels in versatility and robustness, outperforming many existing solutions in diverse network and attack scenarios.
3) Zero-Day Attack Detection: The model's strength lies in detecting zero-day attacks, using an autoencoder that characterises normal traffic and adaptive learning to identify novel threats beyond its training data, enhancing proactive network security.

The paper's organization is as follows: In Section II, we present an overview of related research in the field of anomaly detection using autoencoders. In III, we presented the threat

model. Our proposed approach is detailed in Section IV. Section V presents the results of our experiments. Finally, our conclusions are summarized in Section VI.

II. Related Work

Network data anomaly detection has become a crucial field of research as a result of the increasing difficulties posed by cyberthreats that bypass conventional defences. Anomaly detection, as opposed to systems based on signatures, pinpoints possible dangers by emphasising unexpected deviations.

A. Network Anomaly Detection Techniques

The complexity and size of contemporary networks have increased significantly in the current digital era. This increase demands strict and ongoing monitoring to quickly spot and respond to out-of-the-ordinary tendencies that might indicate possible security breaches or unauthorised intrusions. The academic community and industry specialists have developed several approaches over time to meet these needs:

a) Knowledge-Based Techniques: are rooted in predefined rules and signatures derived from known threat patterns. They represent some of the earliest methods in cybersecurity, with the primary challenge being the creation of pattern recognition techniques that are sufficiently generic yet effective. Techniques like heuristic analysis, signature matching [9], and payload statistical analysis [10] play a crucial role in identifying these patterns. However, their effectiveness is often circumvented by attackers who employ tactics such as embedding noise within the attack vectors to mask their malicious activities. Specifically, they are vulnerable to attacks based on Generative Adversarial Networks (GAN) [11].

Despite their proficiency in detecting and countering known threats, the inherent rigidity of knowledge-based techniques is a significant limitation. This inflexibility leads to ineffectiveness against novel, never-before-seen threat patterns, rendering them unsuitable for defending against modern, sophisticated cyberattacks [12].

b) Statistical-Based Techniques: operate by establishing baselines for typical network behaviors. They have their roots in the field of statistics. When observed data dramatically deviates from these predefined standards, an anomaly is found. Typically, this departure from regular network traffic patterns can be a reliable sign of impending dangers, particularly if there is a significant variance [13].

c) Machine Learning-Based Techniques: enter the field of artificial intelligence and make use of machine learning algorithms. They frequently use clustering techniques to divide network data into groups based on similarity measures. Commonly, each cluster may represent a specific attack when the dataset for training is labelled. This stratification helps identify outliers or abnormalities, alerting us to possible dangers. Particularly in contemporary IoT situations, their flexibility and learning skills have shown great potential in identifying complex abnormalities [14].

B. Exploiting Auto-encoders for Anomaly Detection

The landscape of anomaly detection has been significantly shaped by the advent and evolution of deep learning techniques. Among these, autoencoders have stood out due to their innate ability to model complex non-linear relationships within data. An autoencoder is a neural network designed to reconstruct its input by first encoding it into a compressed representation and then decoding it back to the original form. The projection on the latent space is designed to minimize the loss of information. Thus, the discrepancy between the original input and its reconstruction, termed reconstruction error, becomes a pivotal metric in anomaly detection. Specifically, samples deviating significantly from learned patterns, as evident from a high reconstruction error, are flagged as anomalies [15].

Recent advancements in autoencoder-based techniques for network anomaly detection have shed light on various facets of this challenging domain. Wang *et al.* ventured into a hybrid approach by integrating the BIRCH clustering algorithm with autoencoders. Their amalgamation aimed to ameliorate the computational complexity and bolster detection accuracy, a claim substantiated by their tests on four distinct network security datasets. However, they recognized the scope for refining their algorithm and the challenges posed by limited datasets [16].

Min *et al.* introduce a novel network intrusion detection method using a Memory-Augmented Deep Autoencoder (MemAE). This method addresses the over-generalization problem of traditional autoencoders by incorporating a memory module that learns normal input patterns, thus improving the detection of anomalies. The MemAE model is trained to reconstruct abnormal samples close to normal samples, thereby enhancing the detection of network intrusions [17]. The efficacy of the approach is demonstrated through experiments on the CICIDS2017 dataset [18], which offers a novel solution to handle high data dimensionality in cybersecurity contexts.

Also, Yang *et al.* proposed a network intrusion detection system for Software-defined Networks, utilizing unsupervised machine learning for the real-time detection of both known and zero-day attacks. Griffin's used Kitsune dataset [19] to train and operate a set of autoencoders, achieving high accuracy with reduced complexity and latency [20].

AE-LDA suggests a real-time resilient framework for the detection of both conventional and elusive zero-day network intrusions. This unique integration sets our approach apart, particularly in its proficiency to identify and address both known and unknown network threats effectively.

III. Threat Model and Assumptions

A. Threat Model

In developing our network intrusion detection framework, we have comprehensively considered the spectrum of threats across different network layers:

TABLE I
COMPARISON OF MACHINE LEARNING METHODS FOR ANOMALY DETECTION

Feat. Anal.	ZDA Det.	High Acc.	Robust.	RT Det.
Signature based [9]	✓	✓	×	✓
Payload statistics [10]	✓	✓	×	×
Flow statistics [13]	✓	×	×	×
AE-LDA (ours)	✓	✓	✓	✓

1) Data Link Layer Threats: MAC spoofing, ARP poisoning, and identity falsification are prevalent. Our solution incorporates mechanisms to detect unusual patterns in the MAC address behaviour, which can be indicative of malicious activities such as identity spoofing or unauthorised network access attempts.

2) Network Layer Threats: encompass a variety of attacks, including IP spoofing, routing attacks, and unauthorised packet sniffing. Our approach involves the use of advanced analytics to monitor for irregular traffic flows, suspicious routing patterns, and other signs of intrusion, thereby providing a robust defense against attacks targeting the network layer.

3) Transport Layer Threats: include SYN flooding, session hijacking, and port scanning. Our system employs techniques to recognise abnormal session patterns, unexpected port activities, and other anomalies that could signal security breaches at the transport layer.

4) Application Layer Threats: consider zero-day exploits, advanced malware, and targeted phishing attacks. By utilising deep learning and pattern recognition, our framework analyses application-level data to identify signs of malicious activities, ensuring a high degree of protection against these increasingly sophisticated attacks.

Table I provides a comparative overview of various machine learning methods used for anomaly detection, highlighting their capabilities in terms of feature analysis. In the table, the abbreviations used are: 'Feat. Anal.' for Feature Analysis, 'ZDA Det.' for Zero-Day Attacks Detection, 'High Acc.' for High Accuracy, 'Robust.' for Robustness, and 'RT Det.' for Real-time Detection. This stratified approach to threat modelling allows us to tailor our defence mechanisms to the specific vulnerabilities and attack vectors inherent to each network layer, ensuring a comprehensive and effective network intrusion detection system.

B. Design Goals and Assumptions

Our framework is strategically designed to provide a comprehensive and resilient approach to network intrusion detection, addressing the following key goals and assumptions:

1) Enhancing Network Reliability with Real-time Monitoring: Our system is specifically designed to strengthen network reliability by integrating real-time monitoring capabilities. This approach ensures that the continuous surveillance and intrusion detection activities are efficiently managed, minimizing any impact on network performance. By providing a security solution that operates seamlessly within real-time constraints, we not only safeguard against intrusions but also uphold the integrity and stability of network operations. This real-time aspect is crucial in maintaining network reliability, as it allows for immediate detection and response to potential threats, ensuring uninterrupted and stable network functioning.

2) Explainable AI for Transparent Decision-Making: The integration of Explainable AI (XAI) in our system ensures that the decision-making process is both transparent and understandable. This is essential for network administrators to gain insights into the system's alerts, leading to informed and effective security management.

3) Detection of Unknown and Evolving Threats: We have a strong focus on identifying and mitigating non-conventional, previously unrecognised network threats, including zero-day attacks and novel malware types. By analysing network patterns that deviate from the norm, our system provides a robust defence against these emerging security risks.

4) Resilience Against Evasion Techniques: Recognising the evolving nature of cyber threats, our system is designed to remain effective against sophisticated evasion tactics employed by attackers. This involves maintaining high detection accuracy even as attackers modify their strategies to evade traditional security measures.

By aligning our design goals and assumptions with these key areas, we ensure a robust, adaptable, and efficient framework for network intrusion detection. This holistic approach not only addresses current security challenges but also anticipates future threats, ensuring the long-term resilience and reliability of network systems.

IV. PROPOSED APPROACH FOR AE-LDA

In this section, we present our proposed approach for anomaly detection, which encompasses a comprehensive methodology designed to enhance the accuracy and interpretability of anomaly detection in network security (see Fig. 3). The approach is composed of two main components: i) Feature Extraction and ii) Anomaly Detection Model.

A. Feature Extraction

Our preprocessing pivotally rests on extracting significant features from raw PCAP files (see Fig 1). The primary goals of this feature extraction are twofold: first, to condense the intricate web of network interactions into clear, measurable patterns; and second, to adapt this raw data into a format amenable for machine learning model training. PCAP files offer a holistic view of network traffic. By analyzing these files, we can gain insights into network behaviour, uncover potential vulnerabilities, and detect malicious activities. Our main challenge is to methodically identify and categorize the crucial data contained within these packets. To address this, our approach is centred on devising an efficient, exhaustive set of features apt for further analysis. This section provides an in-depth look at our strategy for extracting features from PCAP files, laying the groundwork for our comprehensive network traffic study. We've divided the extracted features into five key

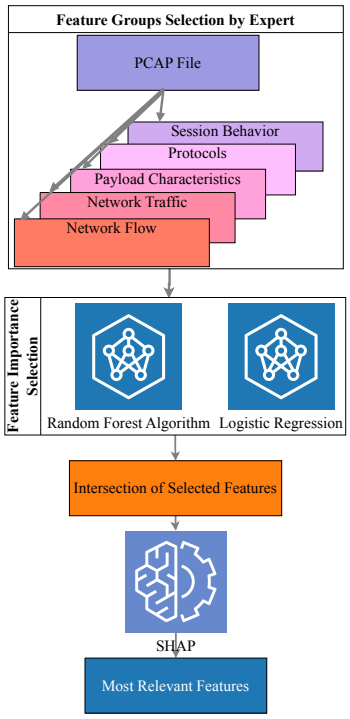

Fig. 1. The process workflow of Feature Selection

categories, each illuminating different facets of network interactions: Network Traffic Features, Session-related Information Features, Network Flow Features, Protocol-specific Features, and Payload Characteristics Features.

Our feature extraction process combines meticulously the robustness of a Random Forest algorithm with the detailed insights of SHapley Additive exPlanations (SHAP). The SHAP value determines how a given feature *explains* (impacts) the model's prediction. We initiate with a Random Forest to identify key features, where the Gini importance of each feature f is calculated as:

$$\text{Gini Importance}(f) = \frac{1}{N}\sum_{i=1}^{N}\text{Impurity Decrease}_i(f) \quad (1)$$

Where N is the number of trees, and $\text{Impurity Decrease}_i(f)$ represents the decrease in impurity in the i-th tree due to feature f.

To further refine and understand the importance of these features, we employ SHAP values. For a feature f, its SHAP value is determined by:

$$\text{SHAP}(f) = \frac{1}{M}\sum_{j=1}^{M}\text{Marginal Contribution}_j(f) \quad (2)$$

Where M is the number of all possible permutations of features, and $\text{Marginal Contribution}_j(f)$ denotes the change in the prediction outcome when including feature f in the j-th permutation.

Algorithm 1 Feature Extraction Algorithm

Train a Random Forest model on the dataset.
for each feature f in the dataset **do**
  Calculate Gini importance of f with eq. 1.
end for
Prune features based on a set importance threshold to reduce model complexity.
for each remaining feature f **do**
  Compute SHAP values for f with eq. 2 to understand its contribution.
end for
Utilize optimized TreeSHAP for large datasets to balance detail and efficiency.

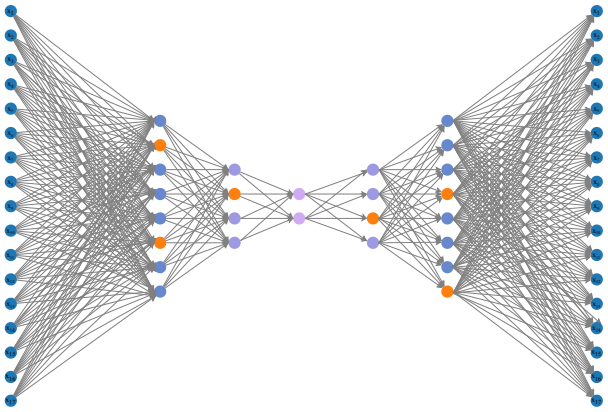

Fig. 2. The autoencoder structure

This comprehensive approach lays a solid foundation for our approach (see Algorithm 1), ensuring it is underpinned by the most informative and relevant features for enhanced predictive accuracy and interpretability.

B. Anomaly Detection Model

Then, we have to detect anomalies in the traffic. We rely on an autoencoder with Linear Discriminant Analysis (LDA) to characterize the usual network traffic, and thus, to detect anomalies.

a) Autoencoder for Anomaly Detection: We train the autoencoder to capture the normal behavior of network traffic. Thus, we train the model with all the data which is i) generated by the network when we are sure that no attack occurs (i.e, at the first stage of the deployment), ii) a training dataset without data labeled with an attack. Very classically, the training objective of the autoencoder is to minimize the Mean Squared Error (MSE) between the input vector x and its reconstruction $\hat{x}$, given by:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2 \quad (3)$$

where n is the number of features. Anomalies are identified when the reconstruction error exceeds a predefined threshold θ.

The specific structure of the autoencoder is described in Fig. 2. The architecture of the autoencoder is a key aspect of its design, reflecting our focus on capturing the intricate patterns within the data. The input layer of the autoencoder consists of 17 neurons, corresponding to the 17 selected features in our scenario. This layer is followed by a series of encoding layers that successively reduce the dimensionality of the input data, capturing the most salient aspects of the traffic patterns. The first encoding layer reduces the input dimensions by half, utilizing 8 neurons, and employs a ReLU activation function. A dropout layer with a 30% dropout rate follows to mitigate the risk of overfitting. The second encoding layer continues this dimensionality reduction, halving the number of neurons to 4, and is similarly followed by a ReLU activation and a dropout layer. We do not employ a (de)convolutional layer since the packet's features are not spatially dependent as pixels of an image are.

The bottleneck, or latent space, of our autoencoder, consisting of 2 neurons, serves as the crux of the model. Here, the data representation is at its most compressed, encapsulating the core characteristics of the normal traffic patterns.

Symmetric to the encoding path, the decoding layers of the autoencoder serve to reconstruct the input data from its compressed form. The first and second decoding layers mirror their encoding counterparts, progressively increasing the dimensions back to the original size (17 features in our case). Each decoding layer is equipped with a ReLU activation function and a dropout layer to maintain consistency and effectiveness in the reconstruction process.

The output layer, with 17 neurons, completes the architecture. It employs a sigmoid activation function to ensure that the output data mirrors the scale of the original input data.

For training, we use an Adam optimizer with a learning rate of 0.001 and a mean squared error loss function to fine-tune the model. To further enhance the model's performance and avoid overfitting, we implement an early stopping mechanism, which halts training if there is no improvement in the validation loss over five consecutive epochs. The model is trained for up to 50 epochs with a batch size of 256, utilizing shuffled mini-batches for each training cycle.

b) LDA for Anomaly Classification: In our approach to network security, detecting anomalies is only the first step; we propose then to classify these anomalies. To achieve this, we employ Linear Discriminant Analysis (LDA), a technique adept at distinguishing between predefined classes in a dataset. In the context of our study, these classes are explicitly constructed using a labelled dataset, which includes various types of network traffic with attacks.

The LDA operates on the premise of finding a linear combination of features that best separates the different classes in the dataset. This separation is crucial, as it allows for a clear distinction between normal network behaviour and various types of attacks, which we label as 'attack X', 'attack Y', etc., alongside the 'normal' traffic label. This distinction is visually and analytically important, as represented in Figure 3, where different types of network traffic are used to train distinct parts of the model.

Algorithm 2 AE-LDA Algorithm

```
Train an autoencoder on normal network traffic data.
for each new data point do
    Calculate the reconstruction error.
    if error exceeds θ then
        Flag the data point as an anomaly.
        Use LDA to classify the anomaly into a specific
        category.
    end if
end for
```

An interesting scenario arises when our autoencoder, designed for anomaly detection, fails to flag an instance as anomalous, but the subsequent classification by the LDA indicates an attack. While the LDA is very robust in detecting known attacks, the autoencoder has a more generic purpose: it aims at characterising normal traffic. Thus, the autoencoder targets rather zero-day attacks, that cannot be detected by LDA because the classification algorithm has not been trained in that way.

This dual approach, combining the anomaly detection capabilities of the autoencoder with the classification power of LDA, provides a comprehensive toolset for network security. While the autoencoder excels at flagging deviations from normal traffic patterns, the LDA offers a nuanced understanding of the type of attack, enabling a more targeted and effective response to potential security threats. The classification is based on the linear combination of features that best separates different classes, using the decision rule:

$$\text{Decision Rule} = \log \frac{P(y|x)}{P(\neg y|x)} \tag{4}$$

where $P(y|x)$ is the probability of the anomaly belonging to a specific class given the features x.

The approach (see Algorithm 2) capitalizes on the capabilities of deep learning for anomaly detection and employs statistical analysis for classification, thereby offering a holistic solution for the monitoring of network security.

V. Experimental Evaluation

In this section, we rigorously assess the performance of our proposed approach across various dimensions. This evaluation is meticulously designed to validate the model's effectiveness in detecting and classifying network anomalies, emphasizing both accuracy and real-time responsiveness. We exploit two different datasets (i.e, CICIDS2017 [21], Kitsune [19]) to illustrate the genericity of our approach and its robustness to detect anomalies in a wide range of network activities and

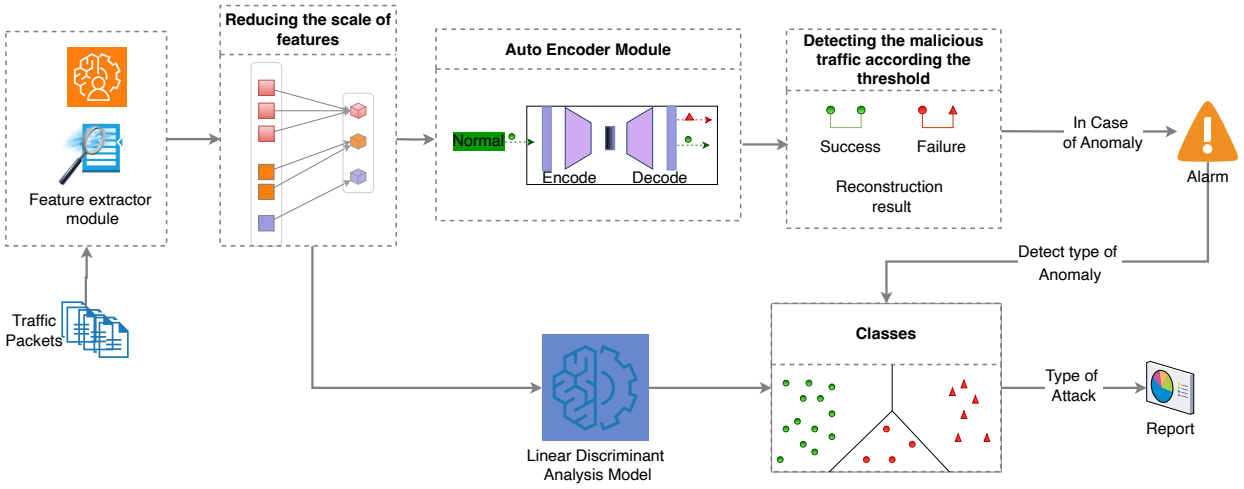

Fig. 3. The process workflow of Anomaly detection

attack scenarios. We measure usual key performance metrics expressed as follows:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

$$Precision = \frac{TP}{TP+FP} \quad (6)$$

$$Recall = \frac{TP}{TP+FN} \quad (7)$$

$$F1-Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (8)$$

$$Prediction\ time = \frac{T_{\text{total}}}{N_{\text{packets}}/N_{\text{packets per flows}}} \quad (9)$$

Where TP, TN, FP, and FN stand for respectively *True Positives*, *True Negatives*, *False Positives* and, *False Negatives*.

A. Preliminary performance evaluation

We first evaluate the ability of our model to detect anomalies and particularly unknown attacks. For this purpose, we train the autoencoder with benign traffic only. For the test, we use a dataset with attacks and benign traffic not used for the training phase. A critical aspect of our evaluation is the model's proficiency in detecting unknown attacks, a feature crucial for real-world applications. This capability is exemplified by an F1-score of 0.9417 and an accuracy of 0.9590 observed in our experiments. Significantly, the model demonstrates a remarkable ability to identify novel threats, indicative of its robustness and adaptability to evolving security challenges. This efficacy in recognizing zero-day attacks underpins the advanced anomaly detection techniques integrated into our model. The ROC curve (see Fig. 4) further illustrates the balance achieved between sensitivity and specificity across various operational thresholds, indicating the model's consistent performance under different conditions.

B. Performance on CICIDS2017 Dataset

We compare the performance of the following approaches:

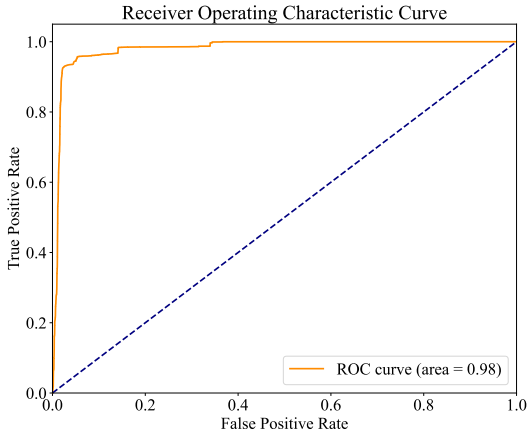

Fig. 4. ROC Curve depicting the model's sensitivity and specificity across varying thresholds.

TABLE II
COMPARISON OF AUROC PERFORMANCE FOR CICIDS2017 FOR DIFFERENT MODELS

Model	AUROC
OCSVM [22]	0.7684
AE [23]	0.8758
MemAE [17]	0.9101
AE-LDA	0.98

- OCSVM [22] relies on a Support Vector Machine (SVM) to classify the traffic in different attacks/benign traffic;
- AE [23] combines an autoencoder with a one-class SVM approach using the latent space;
- MemAE [17] exploits an autoencoder with a memory module ;
- LAE-LDA is our approach described in section IV.

It is worth noting that Griffin [19] didn't provide their implementation. Thus, we are not able to compare LAE-LDA with Griffin using the CICIDS2017 dataset. We provide full access

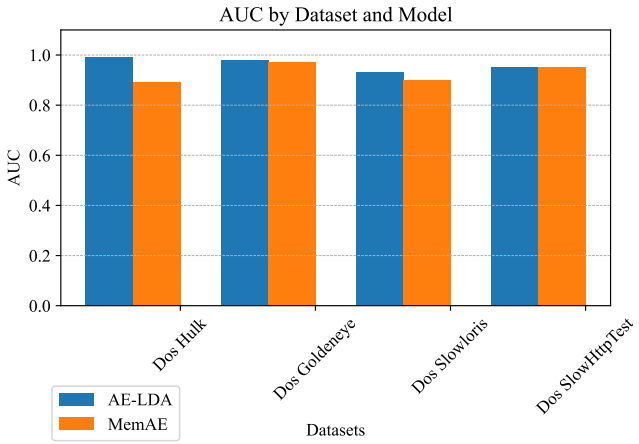

Fig. 5. Comparative ROC Curve Analysis on CICIDS2017 dataset.

TABLE III
DETAILED PERFORMANCE METRICS OF OUR MODEL ON CICIDS2017

DoS Attacks	Accu-racy	Reconstruc. Err. Threshold	Mean Square Error	Area Under Curve	Detection Time
Hulk	0.9811	31.1040	4.1459	0.99	11.99
Goldeneye	0.98	31.0997	1.0230	0.9772	11.93
Slowloris	0.9800	31.2328	0.8357	0.93	11.77
Slow-HttpTest	0.9873	31.1841	0.7761	0.95	12.04

to our code [24] on GitHub to ensure that other researchers and practitioners can validate, reproduce, and build upon our work. This transparency is to our mind crucial in the field of cybersecurity.

We evaluated these models using the Area Under the Receiver Operating Characteristic (AUROC) metric, a crucial indicator of a model's ability to distinguish between different classes (i.e., attacks and benign traffic). Higher AUROC values signify greater discriminative power. As shown in Table II and Fig. 5, LAE-LDA outperforms other models, including our closest competitor, MemAE, across all types of attacks.

Table III provides a detailed breakdown of LAE-LDA's performance for each attack type within the CICIDS2017 dataset. Noteworthy is the model's consistent detection time of under 12 ms, irrespective of the attack type. This constancy is attributed to the fixed number of computational operations required, regardless of traffic volume, a critical feature for real-time intrusion detection systems.

The *Reconstruction Error Threshold* in Table III refers to the maximum deviation from normal traffic patterns that the model tolerates before flagging an anomaly. The *Mean Square Error* (MSE) estimates the model's accuracy in reconstructing input data, with lower MSE indicating higher fidelity. These metrics, combined with high AUROC scores, demonstrate LAE-LDA's robustness and precision in detecting a wide range of attacks.

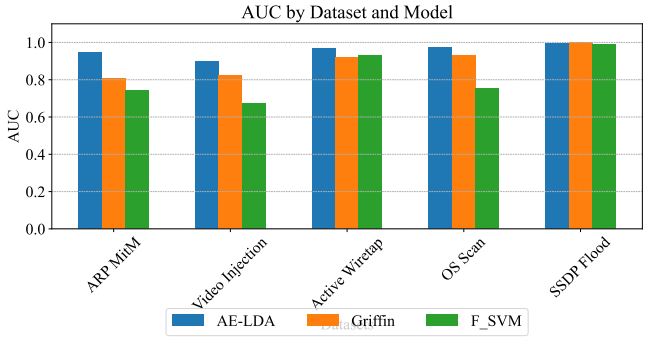

Fig. 6. Performance evaluation on the Kitsune dataset.

TABLE IV
DETECTION ACCURACY COMPARISON ON KITSUNE DATASET

Method	AE-LDA	Griffin	pcStream2	F_SVM	F_RF
ARP MitM	0.9487	0.8048	0.7219	0.7452	0.6512
Video Injection	0.9007	0.8237	0.5816	0.6718	0.6139
Active Wiretap	0.9669	0.9188	0.7413	0.9281	0.7634
OS Scan	0.9713	0.9281	0.7513	0.7517	0.7212
SSDP Flood	0.9945	0.9999	0.9971	0.9876	0.8674

In conclusion, the comprehensive evaluation on the CICIDS2017 dataset underscores the advanced detection capabilities of LAE-LDA. Our approach not only exhibits superior performance in recognizing known attack patterns but also shows promising potential in identifying novel and sophisticated cyber threats. This advancement sets a new benchmark in the IDS domain and opens avenues for further research into adaptive and intelligent cybersecurity solutions.

C. Performance on Kitsune Dataset

We compare LAE-LDA with Griffin [20] and the approaches already included in the original paper using the Kitsune dataset [19]. Since the code of Griffin is not available online, we can only compare our solution with Griffin using the same dataset, extracting their original results directly from their paper.

As depicted in Fig. 6 and Tab. IV, our model, incorporating an autoencoder and LDA, exhibits a strong average detection capability across various network scenarios. Griffin outperforms LAE-LDA only for the detection of the SSD Flood attack, which is also well detected by other competitors.

The integration of LDA with the autoencoder in our model is particularly noteworthy for its efficacy in classifying a wide spectrum of network anomalies, as evidenced by the high AUROC values. The model's ability to maintain consistent performance across a variety of attack scenarios, along with the open availability of our implementation, reinforces the practical applicability and reliability of our approach in real-world security contexts.

D. Discussion

The experimental evaluation of our proposition pinpoints promising results and underscores its potential in the domain of network security. This discussion aims to delve deeper into these findings, exploring their implications and the broader impact of our research.

1) Model's Robust Performance: A key highlight of our model is its robust performance across diverse datasets, particularly in the detection of sophisticated network anomalies. The high accuracy observed with the CICIDS2017 and Kitsune datasets demonstrate the model's effectiveness in handling a variety of attack scenarios, ranging from DoS attacks to more subtle and complex threats like Active Wiretap. This versatility is crucial for practical deployment in real-world network environments, where the nature of threats can be highly varied and unpredictable.

2) Comparative Analysis and Model Superiority: The comparative analysis conducted with existing methodologies, such as OCSVM, AE, and MemAE, reveals a clear superiority of our model in terms of AUROC performance. This superiority is not only a testament to the efficacy of our approach but also highlights the potential shortcomings and areas for improvement in existing methods.

3) Real-time Responsiveness: The prediction time results from our model are indicative of its suitability for real-time application. In the context of network security, where timely response is critical, the ability of our model to provide quick and accurate predictions is a significant advantage. This aspect ensures that our model can be an effective tool in preventing the escalation of network threats in real-time scenarios.

VI. Conclusions

In conclusion, this study introduces a robust and adaptable model for network anomaly detection, which has demonstrated superior performance across diverse datasets, particularly CICIDS2017 and Kitsune. Its effectiveness in accurately detecting a broad spectrum of network anomalies, including zero-day attacks, significantly advances the field of network security.

Looking forward, our research opens up several avenues for future work. Further refinement and optimization of the model to differentiate anomaly from fault diagnosis of network and anomaly caused by malicious activity. Also, trying to predict the attack or anomaly from the behaviour of the network needs further investigation. In particular, we would like to detect an ongoing attack from the very beginning, where small anomalies cumulate to detect malicious behaviours.

Acknowledgements

This research was funded by the Federal Ministry of Education and Research (BMBF) under reference number COSMIC-X 02J21D144, and supervised by Projektträger Karlsruhe (PTKA).

References

[1] Y. Yamout, T. S. Yeasar, S. Iqbal, and M. Zulkernine, "Beyond smart homes: An in-depth analysis of smart aging care system security," *ACM Computing Surveys*, vol. 56, no. 2, pp. 1–35, 2023.

[2] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: methods, systems and tools," *Ieee communications surveys & tutorials*, vol. 16, no. 1, pp. 303–336, 2013.

[3] N. Hubballi and V. Suryanarayanan, "False alarm minimization techniques in signature-based intrusion detection systems: A survey," *Computer Communications*, vol. 49, pp. 1–17, 2014.

[4] C. Kreibich and J. Crowcroft, "Honeycomb: Creating intrusion detection signatures using honeypots," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 1, pp. 51–56, jan 2004.

[5] D. K. Reddy *et al.*, "Deep neural network based anomaly detection in internet of things network traffic tracking for the applications of future smart cities," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 7, p. e4121, 2021.

[6] Y. Imrana, Y. Xiang, L. Ali, and Z. Abdul-Rauf, "A bidirectional lstm deep learning approach for intrusion detection," *Expert Systems with Applications*, vol. 185, p. 115524, 2021.

[7] A. Lazaris and V. K. Prasanna, "An lstm framework for modeling network traffic," in *IFIP/IEEE IM*, 2019, pp. 19–24.

[8] Z. Chen, C. K. Yeo, B. S. Lee, and C. T. Lau, "Autoencoder-based network anomaly detection," in *WTS*. IEEE, 2018.

[9] B. J. Radford, L. M. Apolonio, A. J. Trias, and J. A. Simpson, "Network traffic anomaly detection using recurrent neural networks," *arXiv preprint arXiv:1803.10769*, 2018.

[10] K. Wang and S. J. Stolfo, "Anomalous payload-based network intrusion detection," in *International workshop on recent advances in intrusion detection*. Springer, 2004, pp. 203–222.

[11] R. Chauhan and S. Shah Heydari, "Polymorphic adversarial ddos attack on ids using gan," in *ISNCC*, 2020.

[12] A. Prayote, "Knowledge based anomaly detection," Ph.D. dissertation, UNSW Sydney, 2007.

[13] C.-Y. Lin and S. Nadjm-Tehrani, "Timing patterns and correlations in spontaneous SCADA traffic for anomaly detection," in *RAID*, 2019, pp. 73–88.

[14] A. Chatterjee and B. S. Ahmed, "Iot anomaly detection methods and applications: A survey," *Internet of Things*, vol. 19, p. 100568, 2022.

[15] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in *international conference on knowledge discovery and data mining*. ACM SIGKDD, 2017, pp. 665–674.

[16] D. Wang, M. Nie, and D. Chen, "Bae: Anomaly detection algorithm based on clustering and autoencoder," *Mathematics*, vol. 11, no. 15, p. 3398, 2023.

[17] B. Min, J. Yoo, S. Kim, D. Shin, and D. Shin, "Network anomaly detection using memory-augmented deep autoencoder," *IEEE Access*, vol. 9, pp. 104 695–104 706, 2021.

[18] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization." *ICISSp*, vol. 1, pp. 108–116, 2018.

[19] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: an ensemble of autoencoders for online network intrusion detection," *arXiv preprint arXiv:1802.09089*, vol. https://doi.org/10.24432/C5D90Q, 2018.

[20] L. Yang, Y. Song, S. Gao, A. Hu, and B. Xiao, "Griffin: Real-time network intrusion detection system via ensemble of autoencoder in sdn," *IEEE Transactions on Network and Service Management*, vol. 19, no. 3, pp. 2269–2281, 2022.

[21] "Intrusion detection evaluation dataset (CIC-IDS2017)," https://www.unb.ca/cic/datasets/ids-2017.html.

[22] B. Schölkopf, R. C. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt, "Support vector method for novelty detection," *Advances in neural information processing systems*, vol. 12, 1999.

[23] L. Mhamdi, D. McLernon, F. El-Moussa, S. A. R. Zaidi, M. Ghogho, and T. Tang, "A deep learning approach combining autoencoder with one-class svm for ddos attack detection in sdns," in *ComNet*. IEEE, 2020.

[24] F. Stodt, "AE-LDA," GitHub repository, 2023, [Online; accessed 2-April-2024]. [Online]. Available: https://github.com/f11691/Behaviour_Anomaly_Detector