A Hybrid Autoencoder-GAN Approach for Detecting Zero-Day Attacks in Semi-Labeled Network Traffic

STUDENT NAME: NGUYEN DUC BINH
STUDENT NUMBER: MSE13183

Lecturer: Phan Duy Hung

DATE OF SUBMISSION: 23/4/2025

TABLE OF CONTENTS

INTRODUCTION TO THE RESEARCH PROPOSAL 3
INTRODUCTION.....3
PROBLEM STATEMENT 3
SIGNIFICANCE OF THE STUDY 4
BACKGROUND AND LITERATURE REVIEW 5
BACKGROUND 5
LITERATURE REVIEW 7
RESEARCH OBJECTIVES, METHODOLOGY & RESOURCES...8
RESEARCH OBJECTIVES 8
RESEARCH METHODOLOGY 9
RESOURCES 12
TIMESCALE.....13
REFERENCES.....14

INTRODUCTION TO THE RESEARCH PROPOSAL

INTRODUCTION

In the current landscape of cybersecurity, zero-day attacks—those that are previously unknown or lack signature-based identifiers—are emerging as a serious threat. According to the 2019 McAfee Labs report, "unknown" attack vectors ranked among the top ten most significant threats, indicating that zero-day attacks can cause substantial damage if not detected in time [1]. However, traditional intrusion detection systems (IDS), which mainly rely on signature-based detection, often fail to identify these new forms of attacks due to the absence of matching patterns in their databases.

The urgent need to protect network systems from unseen threats has driven interest in anomaly-based detection methods using machine learning and deep learning. Unlike signature-based approaches, anomaly detection does not search for known attack patterns but instead identifies deviations from normal behavior, making it particularly useful for uncovering new attacks [2]. One major challenge, however, is the lack of fully labeled network data. In real-world scenarios, only a small portion of network traffic is confirmed to be malicious, while the majority remains unlabeled or partially labeled. This raises a key research question: How can deep learning techniques be effectively applied to detect zero-day attacks within semi-labeled network data?

To address this challenge, the present study proposes a novel intrusion detection framework that combines Autoencoder and Generative Adversarial Network (GAN) techniques. Autoencoders are unsupervised neural network models that reconstruct input data and are commonly used for anomaly detection, as abnormal inputs typically produce high reconstruction errors. GANs, on the other hand, are adversarial models capable of generating synthetic data or learning to distinguish real from fake samples, making them promising for enhancing detection accuracy. The integration of Autoencoder and GAN is expected to result in a more robust intrusion detection method, capable of identifying subtle anomalies indicative of zero-day attacks in modern network datasets such as CICIDS2017 and LS'17.

PROBLEM STATEMENT

The central research problem of this study is how to effectively detect zero-day attacks in semi-labeled network traffic. Specifically, we aim to address the question: *How can abnormal network behaviors that have no prior identifiable patterns be accurately recognized, especially when only a small portion of the data is labeled as either malicious or benign?*

Currently, this problem faces several key challenges. Firstly, traditional supervised learning models require fully labeled training data for both attack and normal traffic classes—an assumption that rarely holds for zero-day attacks. Secondly, conventional machine learning models often exhibit a significant drop in performance when

encountering previously unseen attack types. For example, He et al. (2021) reported that the accuracy of traditional classifiers dropped by up to 30% when evaluated on zero-day malware samples absent from training. This highlights that models relying on predefined attack signatures are likely to miss a substantial portion of zero-day intrusions [3].

This leads to the necessity for a novel approach that is not fully dependent on labeled data, yet can learn from unlabeled network traffic and effectively detect anomalies. To address this, our study proposes combining Autoencoders, which are unsupervised models effective in anomaly detection, with Generative Adversarial Networks (GANs), which can generate synthetic samples and learn to distinguish between normal and abnormal traffic, thereby enhancing classification boundaries.

SIGNIFICANCE OF THE STUDY

Scientific Significance: This research contributes to the field of deep learning-based cybersecurity, particularly in the area of intrusion detection systems (IDS). The integration of Autoencoder and Generative Adversarial Networks (GAN) represents a novel approach in anomaly detection. While prior studies have primarily employed Autoencoders or GANs independently—for anomaly detection and data generation respectively—this study aims to build an enhanced hybrid model that offers superior capability in recognizing zero-day attacks. From an academic perspective, the research seeks to explore whether this combination can improve the trade-off between detection rate and false alarms. The expected outcomes will also advance understanding of how generative models can be applied alongside detection models to address the semi-supervised learning challenges in cybersecurity [4]. This is a valuable contribution, especially as recent surveys have emphasized the need for more advanced deep learning methods in IDS research [1], [4].

Practical Significance: If successful, the proposed model will enhance the effectiveness of real-world intrusion detection systems. Specifically, it will have the ability to detect previously unknown attacks early, enabling cybersecurity professionals to respond proactively before zero-day vulnerabilities are exploited. Its capability to learn from unlabeled network traffic allows the system to fully leverage the abundant but mostly unlabeled data in real environments, thereby reducing reliance on manual expert labeling. Moreover, minimizing false alarms is another critical benefit, as it improves detection precision and reduces the workload on security analysts [5]. In the long term, the results of this study could be integrated into existing IDS platforms or serve as a foundation for developing more intelligent and adaptive cybersecurity monitoring solutions, helping organizations defend against increasingly sophisticated cyber threats.

BACKGROUND AND LITERATURE REVIEW

BACKGROUND

Intrusion Detection Systems (IDS): An Intrusion Detection System is designed to monitor network traffic and detect unauthorized access or malicious activities. There are two primary detection approaches:

1. Signature-based detection (misuse detection): This method compares traffic patterns against known attack signatures—such as specific malware fingerprints or byte sequences—to detect intrusions. It is effective for identifying previously known attacks but fails when facing novel threats.
2. Anomaly-based detection: This method builds a model of normal system behavior and flags any deviation from this model as potentially malicious. Based on the assumption that malicious traffic exhibits unusual characteristics, anomaly-based detection can identify previously unseen attacks [2]. However, its main limitation lies in its high false positive rate, as not all anomalies necessarily indicate attacks.

Zero-day Attacks: A zero-day attack refers to the exploitation of a security vulnerability that is unknown to the public or security vendors. These attacks are especially difficult to detect with traditional IDS, which rely on pre-existing signatures. Detection of zero-day threats often depends on identifying abnormal behavioral patterns or utilizing advanced analysis techniques. For example, a zero-day malware might be identified through abnormal hardware-level behaviors [3] or through anomalous traffic patterns. The key challenge is that zero-day samples are unlabeled in the training data, requiring machine learning models to infer threats based on what is considered normal. This study focuses on enhancing that capability through deep learning methods.

Semi-labeled Data: In many real-world scenarios, collected network logs contain only partially labeled data—where some records are clearly identified as malicious or benign, but the majority remain unlabeled. A prominent example is the Locked Shields 2017 (LS'17) dataset, derived from a cyber defense exercise organized by NATO CCDCOE. The LS'17 dataset contains over 15 million network connections, captured through tools such as Bro (now Zeek). While analysts labeled certain IP addresses as malicious, the vast majority of connections remained unclassified [2]. This is typical in real systems, where incidents are only labeled when confirmed via IDS alerts or firewall logs, leaving much of the data uncertain.
Effectively utilizing such semi-labeled data requires semi-supervised learning techniques, which combine unsupervised learning (to leverage unlabeled data) and supervised learning (on the limited labeled portion) to build effective detection models.

Autoencoder: An Autoencoder is a type of neural network consisting of two main components: an encoder and a decoder. The model is trained to compress the input into a latent representation and then reconstruct the output as closely as possible to the original input. The training objective is to minimize the reconstruction error between

the input and output, allowing the Autoencoder to learn important patterns and features in the data. In anomaly detection, the Autoencoder is typically trained on normal network traffic. It learns to accurately reconstruct familiar patterns, while reconstruction errors become significant when faced with unseen or abnormal inputs, such as malicious traffic. By setting a threshold on the reconstruction error, anomalous connections can be flagged with high sensitivity [2]. Several variants of Autoencoders have been applied in IDS. For example, Yu and Yang proposed a dilated convolutional autoencoder (DCAE) to extract complex network flow features, achieving high classification accuracy and F1-scores of around 98% across various attack types [5]. Due to its unsupervised learning capability, the Autoencoder is a promising component for detecting zero-day attacks.

Generative Adversarial Network (GAN): A GAN is a model composed of two competing neural networks: the Generator, which attempts to produce synthetic data resembling real samples, and the Discriminator, which tries to distinguish between real and generated data. These two networks are trained adversarially: the generator improves to fool the discriminator, while the discriminator improves to better identify fakes. Eventually, the generator learns to create data that matches the distribution of the training set. While widely used in computer vision (e.g., generating photorealistic images), GANs have also shown promise in cybersecurity applications. Two primary use cases include:

1. Synthetic network traffic generation, where GANs create fake attack traffic to augment training datasets.
2. Anomaly detection, where models like GANomaly or ALAD (Adversarially Learned Anomaly Detection) use an Autoencoder-like structure within the generator and leverage the discriminator to evaluate reconstruction quality and detect outliers.

In the study by Yin et al., GANs were used to simulate botnet traffic, which improved botnet detection models by reducing the false positive rate from 19.19% to 15.59% [4]. This demonstrates how GANs can help models focus on difficult-to-separate boundaries, enhancing overall detection performance.

CICIDS2017 Dataset: CICIDS2017 is a benchmark dataset for evaluating IDS performance, created by the Canadian Institute for Cybersecurity (CIC). It simulates one week of enterprise network activity, incorporating a mix of normal traffic and multiple attack scenarios such as DoS, scanning, web-based intrusions, botnets, and brute-force attempts. The dataset provides fully labeled connections, specifying whether each is normal or belongs to a specific attack type. It was designed to replace outdated datasets like KDD99 and NSL-KDD by reflecting realistic network traffic circa 2017. As such, CICIDS2017 is widely used to train and benchmark modern IDS solutions. In this study, CICIDS2017 will be used for both training and evaluating the proposed model. Additionally, parts of the dataset may be converted into a semi-labeled format (e.g., by removing labels for certain attack types) to simulate zero-day attack conditions, thus testing the model's capability in real-world detection scenarios.

LITERATURE REVIEW

The dilated convolutional autoencoder (DCAE) proposed by Yu and Yang has demonstrated the ability to effectively learn from unlabeled network flows, combining the strengths of deep autoencoders and convolutional neural networks. This approach achieved high classification performance with an F1-score of approximately 98% when applied to complex datasets involving botnet and web malware traffic [5]. These results confirm the strong potential of using autoencoders and deep learning to extract hidden patterns in large-scale network data.

In parallel, the feasibility of anomaly detection on semi-labeled datasets has been validated through the work of Klein. In their study, an unsupervised autoencoder model was applied to the LS'17 dataset, which contains only partial labeling (i.e., a subset of connections labeled as attacks). The results showed that the autoencoder was capable of detecting the vast majority of actual attacks in the data, with a recall of 98.2%, including previously unseen (zero-day) attacks. Remarkably, out of 500 flagged anomalies, 50 were confirmed by experts to be genuine zero-day attacks. This finding underscores the value of deep learning models in uncovering latent threats within unlabeled traffic. However, the study also identified a limitation: the model produced a large number of false positives, with precision as low as ~14.2%, meaning that many flagged anomalies were in fact benign. Conversely, the Gradient Boosting Machine model used in the same study achieved 100% precision but suffered from low recall (~7.3%), failing to identify most of the new attacks [2]. This contrast reveals a critical gap: there is a need to enhance anomaly detection models to reduce false positives while maintaining strong zero-day detection capabilities.

To address such limitations, recent research has explored the integration of multiple techniques to leverage the strengths of each. Macas et al. (2020) conducted a comprehensive survey of deep learning methods for IDS and proposed a hybrid architecture. Their model employed stacked non-symmetric deep autoencoders (NDAE) to extract network traffic features, followed by a Random Forest (RF) classifier trained on the learned representations. Comparative results showed that the NDAE+RF model improved classification accuracy by about 5% over the traditional Deep Belief Network (DBN) and also reduced training time by 98.81% [4]. These findings suggest that autoencoders can be effectively combined with powerful classifiers to harness the benefits of both unsupervised learning and supervised classification.

Furthermore, Macas et al. (2020) emphasize the importance of generative deep learning techniques, such as Autoencoders and GANs, in the context of massive unlabeled IoT data. By learning in a "human-like" manner—that is, extracting features directly from raw data—these models can discover anomalous patterns without the need for labels [4]. The authors cited a study in the critical infrastructure domain, where a deep autoencoder was used in a fully unsupervised setting to detect anomalies, achieving high accuracy and fast convergence. In addition, Macas et al. referenced Yin et al., who used GANs to generate synthetic network traffic to improve botnet detection. Their results showed a reduction in false positive rate from 19.19% to 15.59% after incorporating GAN-generated data [4] This clearly demonstrates that GANs can enhance dataset diversity and balance, which is especially beneficial when malicious traffic is rare and imbalanced.

In another approach aimed at detecting previously unknown attacks, He et al. (2021) applied ensemble learning techniques in the context of hardware-level malware detection. Although not directly related to network traffic, their study illustrates the effectiveness of combining multiple models. Specifically, they used AdaBoost to aggregate several weak learners into a stronger one, enhancing the performance of a Random Forest classifier. Their method achieved an F1-score of 92% and a True Positive Rate (TPR) of 95% on a zero-day malware dataset—significantly outperforming individual models [3]. These findings suggest that combining machine learning techniques, whether through ensemble learning or hybrid generative-discriminative models, is key to improving the detection of emerging threats.

In conclusion, prior research supports the following insights:
(i) Autoencoders are effective in identifying anomalies and previously unseen attacks, though they still face issues with low precision;
(ii) GANs and other generative approaches can help reduce false positives and enrich training data;
(iii) Model integration—in various forms—represents a promising strategy for enhancing IDS performance.

Building upon these foundations, this study proposes to extend and integrate these approaches into a unified model for accurately detecting zero-day attacks in semi-labeled network datasets.

RESEARCH OBJECTIVES, METHODOLOGY & RESOURCES

RESEARCH OBJECTIVES

The overarching goal of this research is to develop a novel intrusion detection method that combines Autoencoder and Generative Adversarial Networks (GAN) to effectively identify previously unseen (zero-day) attacks within semi-labeled network datasets. To achieve this, the study is guided by the following specific objectives:

- Objective 1: Conduct a comprehensive review and identify research gaps – Analyze existing intrusion detection approaches related to Autoencoders, GANs, and zero-day attack detection. Identify current limitations such as high false alarm rates and strong reliance on labeled data, which the proposed research aims to address.
- Objective 2: Propose a hybrid Autoencoder-GAN model – Design a novel architecture where the Autoencoder and GAN operate in tandem. Specifically, the Autoencoder is used to detect anomalies, while the GAN supports the model by generating synthetic data or enhancing the distinction between normal and abnormal traffic, thereby improving sensitivity and accuracy in detecting zero-day attacks.

- Objective 3: Implement and train the model on CICIDS2017 and LS'17 datasets – Collect and preprocess both benchmark datasets. Evaluate the proposed model in two key scenarios:
  (i) using CICIDS2017 (fully labeled) to simulate detection of zero-day attacks by withholding labels for some attack types, and
  (ii) applying the model to LS'17 (semi-labeled) to test its robustness in realistic and incomplete labeling conditions.
- Objective 4: Evaluate model performance and benchmark against existing methods – Assess the proposed system using metrics such as True Positive Rate (TPR/Recall), False Positive Rate (FPR), Precision, F1-score, etc. Compare its effectiveness with baseline methods including standalone Autoencoder, standalone GAN, and traditional supervised classifiers trained with partial labels. This comparison will determine the extent of improvement the hybrid model offers.
- Objective 5: Refine the model and propose real-world applications – Based on evaluation results, fine-tune model parameters or architecture to maximize performance. Analyze the model's effectiveness across different types of zero-day attacks and deployment conditions. Suggest potential integration of the model into real-world IDS platforms or future extensions (e.g., adaptation for IoT environments, architectural enhancements using GAN or Autoencoder variants).

RESEARCH METHODOLOGY

To achieve the stated objectives, the study will adopt an experimental design with the following specific steps and methodologies:

1. Data collection and preparation: Download the CICIDS2017 dataset from the CIC institute (including PCAP files or feature-extracted CSV files) and the LS'17 dataset (from the CCDCOE repository or related research teams). Data preprocessing includes: format conversion if necessary (e.g., PCAP to CSV), data cleaning (removal of invalid records), and normalization of input features (e.g., normalization to the [0,1] range or Gaussian normalization for attributes such as byte count, duration, etc.). For CICIDS2017, a subset of features related to network flows can be selected to match LS'17 (as LS'17 mainly contains connection logs). For LS'17, apply partial labeling based on the list of known malicious IP addresses: connections with source or destination IPs in the blacklist will be labeled as "malicious", and the remaining connections will be treated as "unknown" (and handled as normal data during anomaly detection training).
2. Autoencoder model design: Select a suitable autoencoder architecture for network data. For example, use a multi-layer perceptron (MLP) architecture with decreasing hidden layers, or a 1-D convolutional autoencoder (if the data is treated as time series). Determine the size of the latent representation (latent vector) and the activation functions. Train the autoencoder primarily on data considered normal (i.e., the entire dataset excluding known attack samples). The training objective is to minimize the reconstruction error (using Mean Squared Error – MSE as the loss function). After training, define an anomaly

threshold based on the distribution of reconstruction errors on the training set (e.g., the threshold at the 95th percentile) to identify which samples are considered anomalous.

3. GAN model design: Two integration approaches will be experimented with:

- Approach A – GAN for synthetic attack generation: Use labeled attack samples (from CICIDS2017 or the known portion of LS'17) to train a GAN that generates synthetic malicious traffic. Specifically, build a generator (e.g., an MLP network) to produce network feature vectors resembling attack samples, and a discriminator to differentiate between real and fake attack samples. After training, use the generator to create additional synthetic attack data to augment the training set for the autoencoder or an auxiliary classification model. The goal is to help the model better learn the decision boundary between normal and malicious traffic, thereby reducing false alarms.
- Approach B – GAN for anomaly detection (GANomaly): In this approach, the autoencoder is embedded within the GAN architecture. Specifically, the generator is built as an encoder-decoder to reconstruct the input data, while the discriminator receives both real data and reconstructions from the generator and learns to distinguish between them. The generator attempts to reconstruct in a way that makes the discriminator unable to tell the difference. Once training is complete, for any new input sample, we assess the discriminator's ability to distinguish: if the discriminator can easily detect that the sample "does not resemble" the learned normal data (i.e., identifies it as fake due to poor reconstruction), then the sample is likely anomalous. This approach allows the discriminator to serve as an additional anomaly detector, complementing the reconstruction error metric.

In this research, we will experiment with both approaches A and B, then evaluate which is more effective, or even combine both approaches (e.g., using Approach A first to enrich training data for Approach B).

4. Training the hybrid model:
   For Approach A, the training process can be iterative: train the GAN on real attack samples; use the generator to create synthetic data; augment the training set for the autoencoder or classifier; retrain and repeat until performance stabilizes. For Approach B, the generator (autoencoder) and discriminator are trained simultaneously using a GAN loss function (a combination of reconstruction loss and adversarial loss). Advanced training strategies will be applied: alternating optimization of the generator and discriminator for a set number of epochs. Anti–mode collapse techniques will be implemented as needed (e.g., adding noise, mini-batch discrimination).The entire training process will be conducted on a GPU-equipped workstation to accelerate deep learning model computation, using either TensorFlow or PyTorch libraries.
5. Model evaluation and fine-tuning: Once the model is trained, evaluation will be conducted on the test set using the following steps:
    1. Use the labeled portion of CICIDS2017 for evaluation: Attack samples—especially those representing unseen attack types during training—will be tested to verify whether the model flags them as anomalies. Performance metrics such as True Positive (TP), False

Positive (FP), False Negative (FN), and True Negative (TN) will be computed to derive Precision, Recall, F1-score, and False Positive Rate (FPR).

2. For LS'17, where most zero-day attacks are unlabeled, indirect evaluation will be applied:

   - First, verify whether the model can correctly identify connections involving known malicious IPs—this evaluates detection of known attacks.

   - Then, extract a subset of connections flagged as anomalous by the model and request expert review (or compare with the findings in Klein et al., 2018) to estimate how many are likely to be true zero-day attacks.

   - Alternatively, perform cross-dataset evaluation—e.g., train on CICIDS2017 (with certain attack types excluded) and test on LS'17 to examine if the model can detect suspicious activity in LS'17.

3. Compare with baseline models: Implement a standalone autoencoder (without GAN) and a supervised classification model (e.g., Random Forest or SVM using the available labels) as baselines. Compare differences in recall for novel attacks, precision, and F1-score.
4. Document results and analyze outcomes:

   - If the hybrid model shows a significant increase in recall without a substantial drop in precision compared to the standalone autoencoder, it is considered a success.

   - Conversely, if precision improves while recall slightly drops, it may still be acceptable depending on the prioritized objective (in cybersecurity, maximizing recall is typically preferred to avoid missing potential threats, even at the cost of some false positives).

The above research methodology follows an iterative experimentation approach: we will conduct multiple small-scale experiments to fine-tune the model. All processing steps and intermediate results will be carefully recorded for analysis. In addition, research ethics and data integrity will be strictly maintained: only publicly available datasets will be used; the implementation code will be version-controlled and can be published to ensure transparency and reproducibility of results.

RESOURCES

To conduct this research, the following resources will be mobilized:

- Data: Two primary datasets will be utilized: CICIDS2017 and LS'17. The CICIDS2017 dataset is publicly available from the Canadian Institute for Cybersecurity (CIC) and includes tens of gigabytes of PCAP files and extracted feature CSVs. The LS'17 dataset can be acquired from the CCDCOE repository or by contacting the original research team (Klein et al., 2018); this dataset is in Bro log format. In addition, small-scale sample datasets (e.g., NSL-KDD) may be used for initial testing and validation of basic models before applying to more complex traffic.
- Hardware: A high-performance computing environment with GPU support is required to train deep learning models. Specifically, a system equipped with NVIDIA GPU (8GB memory or higher)—such as GTX 1080, RTX 2080, or above—is recommended for efficient training of Autoencoder and GAN models on large-scale data. At least 16GB of RAM is required for data processing, and ~100GB of storage is needed for raw data and results. If available, the use of HPC servers or cloud computing services (e.g., Google Colab, AWS) can significantly accelerate experimentation.
- Software: The research will be implemented in the Python ecosystem for machine learning and deep learning. This includes:
    - Python 3.8+
    - Deep learning frameworks: TensorFlow 2.x or PyTorch
    - Data processing libraries: pandas, NumPy
    - Machine learning utilities: scikit-learn for baseline models and metrics
    - Development tools: Jupyter Notebook, PyCharm, or VSCode

The preferred operating system for development is Ubuntu Linux 20.04, which provides compatibility and ease of installation for deep learning libraries.

- Human Resources: A research student or research team will be responsible for data collection, model development, and result analysis. A supervising academic advisor will provide methodological guidance and evaluate research progress. Where possible, cybersecurity experts may be consulted to verify flagged anomalies—especially in the LS'17 dataset where expert validation is valuable. However, most evaluations will be carried out using automated and quantitative methods.
- Additional Resources:
    - A collection of relevant academic publications will be used for benchmarking and methodological comparison.
    - Visualization tools such as Matplotlib and Seaborn will be used to present results clearly (e.g., ROC curves, reconstruction error plots).
    - Version control systems (e.g., Git) will be used to manage code and research iterations.

All of the above resources are readily available or easily accessible, ensuring that the research can be carried out efficiently within the defined scope.

TIMESCALE

The proposed research is expected to be conducted over a period of **12 months**, with the main milestones outlined below. The timeline may be adjusted flexibly depending on actual results, but overall, the project will adhere to the following general plan:

Time (Month)	Main mission
1 - 2	Conduct a literature review and collect relevant studies. Finalize the detailed research proposal. Establish a solid theoretical foundation on **autoencoders**, **GANs**, and existing **intrusion detection methods**.
3 - 4	**Data preparation:** Download the CICIDS2017 and LS'17 datasets, perform preprocessing and feature analysis. Select appropriate tools and set up the programming environment (installing required libraries). Begin initial experiments by training a basic autoencoder on sample data to validate the workflow.
5 - 6	**Model development:** Design an appropriate Autoencoder architecture and train it on the preprocessed data. Subsequently, design the GAN model (following Approach A and/or B as previously described). Conduct initial training experiments with the standalone GAN model (e.g., training the GAN to generate attack samples or simulating the GANomaly structure on a subset of the data) to ensure the model functions as expected.
7 - 8	**Integration and training of the hybrid model:** Integrate the Autoencoder and GAN based on the optimal approach identified in the previous phase. Train the combined model on the entire training dataset. Perform hyperparameter tuning (e.g., learning rate, batch size, hidden layer architecture) through multiple trial-and-error experiments to achieve the best performance, based on validation set results.
9	**Model evaluation:** Run the model on the CICIDS2017 test set to collect performance metrics (e.g., Precision, Recall, F1-score, FPR, etc.). Apply the model to the LS'17 dataset to identify suspicious connections; compare with available labels (if any) and document notable findings (e.g., which samples were flagged as anomalies). Compare the performance against baseline methods (standalone Autoencoder, supervised learning models) by training and evaluating those baselines on the same dataset.
10	**Result analysis:** Summarize and compare the obtained results. Evaluate the extent to which the research objectives have been achieved. Identify the strengths and weaknesses of the proposed model. If the results do not meet expectations, consider revisiting the model for adjustments or collecting additional data (if feasible), and repeat the training process as necessary.
11	**Finalizing the research report:** Write the main chapters of the thesis/research report, including: Introduction, Theoretical Background, Methodology, Results, and Discussion. Ensure that all references are properly cited. In addition, prepare supporting materials such as tables, figures (e.g., ROC curves, reconstruction error distributions, model architecture diagrams) to effectively illustrate the findings in the report.
12	**Final review and evaluation:** Review the entire report to ensure logical

Time (Month)	Main mission
	consistency and accuracy. Prepare a presentation (if a thesis defense is required). Proceed with the defense or submit the completed research report on time.

It should be noted that throughout the research process, **risk management and schedule adjustments** will be implemented if any issues arise. For instance, if there are difficulties in obtaining or preprocessing the LS'17 dataset, more focus will be placed on CICIDS2017. Similarly, if the GAN model encounters training challenges (e.g., mode collapse), alternative solutions such as a **Variational Autoencoder** may be considered. This plan is designed to ensure that sufficient time is allocated for both model development and result analysis, thereby enabling the project to achieve its stated objectives.

REFERENCES

[1] D. Gümüşbaş, T. Yıldırım, A. Genovese, and F. Scotti, "A comprehensive survey of databases and deep learning methods for cybersecurity and intrusion detection systems," *IEEE Syst J*, vol. 15, no. 2, pp. 1717–1731, Jun. 2021, doi: 10.1109/JSYST.2020.2992966.

[2] J. Klein, S. Bhulai, M. Hoogendoorn, R. Van Der Mei, and R. Hinfelaar, "Detecting Network Intrusion beyond 1999: Applying Machine Learning Techniques to a Partially Labeled Cybersecurity Dataset," *Proceedings - 2018 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2018*, pp. 784–787, Jan. 2019, doi: 10.1109/WI.2018.00017.

[3] Z. He, T. Miari, H. M. Makrani, M. Aliasgari, H. Homayoun, and H. Sayadi, "When machine learning meets hardware cybersecurity: Delving into accurate zero-day malware detection," *Proceedings - International Symposium on Quality Electronic Design, ISQED*, vol. 2021-April, pp. 85–90, Apr. 2021, doi: 10.1109/ISQED51717.2021.9424330.

[4] M. Macas and C. Wu, "Review: Deep Learning Methods for Cybersecurity and Intrusion Detection Systems," *Proceedings - 2020 IEEE Latin-American Conference on Communications, LATINCOM 2020*, Nov. 2020, doi: 10.1109/LATINCOM50620.2020.9282324.

[5] Y. Xin *et al.*, "Machine Learning and Deep Learning Methods for Cybersecurity," *IEEE Access*, vol. 6, pp. 35365–35381, May 2018, doi: 10.1109/ACCESS.2018.2836950.