A Semi-Supervised Ensemble Framework for IoT Network Anomaly Detection

Mina Erfan
University of Ottawa
Ottawa, Canada
merfa006@uottawa.ca

Hesam Nasiri
University of Ottawa
Ottawa, Canada
mnasi101@uottawa.ca

Sahil Khokhar
University of Ottawa
Ottawa, Canada
skhok015@uottawa.ca

Sadra Teymourian
University of Ottawa
Ottawa, Canada
sadra.teymourian@uottawa.ca

***Abstract*—Modern IoT networks face increasingly sophisticated cyberattacks, yet traditional anomaly detection methods depend heavily on large amounts of labeled data that are both difficult and expensive to obtain. In this paper, we introduce a semi-supervised ensemble framework that overcomes this limitation by effectively leveraging both labeled and unlabeled network traffic data. Our approach employs self-supervised learning to extract rich feature embeddings from unlabeled data, which are then combined with predictions from a supervised classifier using a soft voting mechanism. This integration reduces the dependency on manual labeling while enhancing the overall detection performance. Evaluated on the comprehensive ToN-IoT dataset under both anomaly-majority and normal-majority scenarios, our ensemble demonstrates improvements in F1-Score and ROC-AUC compared to individual models. The results underscore the potential of our framework to provide scalable, adaptive, and robust cybersecurity solutions for IoT environments.**

***Index Terms*—Anomaly detection, Semi-supervised learning, Self-supervised learning, Ensemble Learning**

I. Introduction

Modern organizations rely on large-scale, interconnected networks that include everything from traditional IT systems to Internet of Things (IoT) devices. While this connectivity brings efficiency and innovation, it also expands the attack surface, making networks more vulnerable to cyber threats. Hackers, malware, botnets, and zero-day exploits are constantly evolving, finding new ways to bypass security defenses. As a result, network anomaly detection—the ability to spot unusual or suspicious activity in network traffic—has become a critical component of modern cybersecurity.

However, building effective anomaly detection systems comes with a major challenge: the lack of labeled data. To train a machine learning model, you typically need a large dataset where each traffic instance is labeled as either normal or malicious. However labeling network data is slow, expensive, and requires cybersecurity expertise [1]. On top of that, cyber threats are constantly evolving, meaning that a model trained today might be outdated in a few months due to 'concept drift', when attack patterns change over time, making old models unreliable. As a result, traditional supervised machine learning approaches, which rely heavily on labeled data, often fail to detect new and emerging threats without frequent retraining.

To overcome the challenges of data labeling, semi-supervised learning (SSL) methods [2] and self-supervised learning [3] have been proposed, leveraging both labeled and unlabeled data to improve detection accuracy. While these approaches offer promising results, they come with their own set of limitations. Many semi-supervised models rely on assumptions about how labels can be propagated, making them sensitive to noisy or imbalanced datasets [4], [5]. Meanwhile, self-supervised learning methods excel at extracting meaningful representations from unlabeled data but often struggle to integrate these features effectively into a classification model. Additionally, both methods tend to rely on single-model architectures, making them vulnerable to overfitting specific types of attacks, reducing their generalization ability in real-world deployments.

A. Research Gap and Limitations of Existing Approaches

The increasing complexity of IoT networks and the ever-evolving nature of cyberattacks have underscored the limitations of traditional anomaly detection methods, which rely on vast amounts of labeled data that are often impractical to obtain. Although semi-supervised and self-supervised learning techniques have made strides in mitigating the need for extensive manual labeling, most existing approaches remain confined to a single learning paradigm. Semi-supervised methods typically depend on label propagation assumptions that can falter in imbalanced scenarios, while self-supervised techniques excel at learning rich feature representations but often struggle to translate these embeddings directly into high-fidelity anomaly classifiers.

This situation reveals a critical research gap: there is a need for an integrated framework that unifies the strengths of self-supervised representation learning with the robust decision-making capabilities of supervised classification. Our work directly addresses this gap by proposing a novel self-supervised ensemble framework for IoT network anomaly de-

tection. In our approach, we employ a self-supervised learner to extract high-quality embeddings from unlabeled network traffic, thereby reducing the dependency on labor-intensive data annotation. These embeddings are then combined with the predictions from a high-performing supervised model using a soft voting mechanism. This fusion allows the ensemble to harness the complementary strengths of both paradigms, resulting in improved detection performance and enhanced adaptability across different operational scenarios.

Our key contributions are threefold.

1) First, we bridge the gap between self-supervised and supervised methods, yielding a unified framework that significantly enhances anomaly detection accuracy while minimizing the need for labeled data.
2) Second, our ensemble strategy—by leveraging soft voting—effectively integrates confidence information from both models, thereby increasing robustness against varying network conditions and attack patterns.
3) Third, extensive evaluation on the heterogeneous ToN-IoT dataset demonstrates that our framework not only performs well in both anomaly-majority and normal-majority environments but also offers a scalable and adaptive solution for real-world IoT network security challenges.

The remainder of this paper is organized as follows. In Section II, we review related works within the broader literature. Section III describes the ToN-IoT dataset and outlines the experimental setup employed in our study. In Section IV, we detail our proposed methodology, including our supervised and self-supervised approaches and ensemble integration via soft voting. Section V presents our experimental results and a thorough discussion of the findings. Finally, Section VI concludes the paper and offers potential directions for future research.

II. Related Works

The challenge of network anomaly detection, particularly due to the limited availability of labeled data, has led to the development of machine learning approaches that leverage both labeled and unlabeled data. Semi-supervised learning (SSL) methods have been widely explored in this context, as they enhance model accuracy without requiring extensive labeled datasets. Traditional SSL techniques include label propagation, self-training, and hybrid supervised-unsupervised models. More recently, self-supervised learning has emerged as a complementary approach, focusing on learning meaningful feature representations without explicit labels. This section reviews both semi-supervised and self-supervised methods, highlighting key contributions and their limitations.

A. Semi-Supervised Learning Approaches

Semi-supervised learning has been widely applied in network anomaly detection, leveraging a small amount of labeled data along with a larger corpus of unlabeled data to improve model generalization. Traditional semi-supervised techniques include label propagation, co-training, and self-training, each employing different strategies to utilize unlabeled data effectively. Label propagation [6] propagates labels from labeled to unlabeled data based on similarity, assuming that points with similar features share the same label. Co-training [7] uses multiple models trained on different feature subsets, enabling them to iteratively label unlabeled instances for each other. Self-training [8] refines models by using their own predictions as pseudo-labels, allowing them to learn iteratively from both labeled and unlabeled data.

While these methods have demonstrated success, researchers have developed more advanced deep learning-based approaches to further improve network anomaly detection. For instance, Abdel-Basset et al. [5] proposed a semi-supervised deep learning model for intrusion detection in IoT networks. Their model incorporates a Multiscale Residual Temporal Convolutional (MS-Res) module to capture temporal dependencies in network traffic and a Traffic Attention (TA) mechanism to focus on key features. To enhance learning from unlabeled data, a hierarchical training strategy was employed, ensuring that the model gradually learns from network traffic while maintaining temporal order. Tested on CIC-IDS2017 and CIC-IDS2018 datasets, their approach achieved over 99% accuracy in binary classification and demonstrated strong detection performance across multiple attack types. However, the high computational cost may limit its feasibility for real-time deployment.

To address the limitations of traditional label assignment in semi-supervised learning, Wenjuan Li et al. [9] introduced a disagreement-based voting strategy. Instead of relying solely on majority voting, their method identified disagreement among three classifiers. When two classifiers agreed on a label and the third disagreed, the majority vote was assumed to be correct, and the disagreeing classifier was retrained on these instances. This approach reduced the error rate to 15% on the DARPA dataset and 7.3% on a custom industry dataset. However, due to the lack of large-scale evaluation, its generalizability to diverse real-world scenarios remains uncertain.

Further advancing semi-supervised anomaly detection, Weijian Song et al. [10] explored a graph-based approach for time-series data. They transformed network traffic into graph representations and applied a Graph Convolutional Network (GCN) to learn relationships between graph nodes and edges. Their framework employed teacher-student models, where the student model was trained on both labeled and unlabeled data, while the teacher model generated pseudo-labels to guide learning. This method was evaluated on SWaT, WADI, and PSM datasets, achieving F1 scores of 0.89, 0.863, and 0.963, respectively. However, the authors highlighted concerns about model robustness against adversarial attacks, suggesting the need for further security investigation.

In addition to these advancements, Deep SAD (Deep Semi-Supervised Anomaly Detection) [11] has emerged as a generalizable deep learning-based semi-supervised method for anomaly detection. Unlike conventional one-class classification methods, which assume that all unlabeled data is normal,

Deep SAD introduces a balanced entropy framework that maps normal samples into a compact latent space while forcing anomalies into a dispersed representation. The method extends Deep SVDD and has been evaluated on benchmark datasets such as MNIST, Fashion-MNIST, and CIFAR-10, demonstrating performance improvements over traditional semi-supervised approaches. Although the method has not been applied directly to IoT network intrusion datasets, it provides a strong theoretical foundation for designing semi-supervised models that leverage self-supervised pretraining before performing anomaly detection.

To overcome challenges in semi-supervised approaches such as error accumulation and sensitivity to noisy data, researchers have explored hybrid approaches that integrate supervised and unsupervised methods like clustering. These methods typically begin by training a classifier on labeled data and then refine decision boundaries or assign pseudo-labels to unlabeled samples using techniques such as K-means clustering or anomaly detection. By combining both learning paradigms, hybrid models enhance generalization and robustness in real-world network security applications. For instance, Ravi and Shalinie [4] proposed a semi-supervised deep learning approach that integrates a deep feedforward neural network (DFNN) with K-means clustering. The DFNN is trained on labeled samples, while K-means clusters the remaining unlabeled data. To enhance clustering accuracy and minimize computational overhead, the repeated random sampling technique is employed. The model, tested on the NSL-KDD dataset, achieved 99.78% precision, surpassing traditional supervised models. However, its reliance on manual feature selection and inherent clustering assumptions may hinder adaptability to evolving attack patterns in dynamic, real-world environments.

Overall, these advancements demonstrate how semi-supervised learning has evolved from classical methods to deep learning-based approaches that integrate temporal analysis, voting strategies, and graph-based representations. While these methods improve performance, challenges such as computational efficiency, adaptability, and robustness to adversarial threats remain areas for future explore.

B. Self-Supervised Learning Approaches

Unlike conventional semi-supervised methods, self-supervised learning has emerged as a promising approach for anomaly detection, particularly in cases where labeled data is scarce. Self-supervised learning involves training a model to generate useful feature representations by predicting parts of the data from other parts, without relying on any labels. This is achieved by learning unsupervised representations (often called embeddings) that capture complex patterns in the data. These embeddings can then be used for downstream tasks, such as anomaly detection, classification, or clustering.

Aktar and Nur [12] proposed a self-supervised anomaly detection model combining Contractive Autoencoders (CAE) for feature extraction with Deep Support Vector Data Description (DSVDD) for identifying anomalies in IoT networks. Trained exclusively on normal traffic, the model maps normal samples into a hypersphere, flagging deviations as anomalies. Evaluated on the ToN-IoT and IoTID20 datasets, it achieved 99.57% and 99.64% accuracy, respectively. However, its one-class learning approach may struggle to detect attacks that closely resemble normal traffic, and its high computational cost limits real-time deployment in resource-constrained environments. To enhance effectiveness in low-data scenarios, Dina et al. [13] introduced FS3, a hybrid framework that integrates self-supervised with few-shot learning (FSL) to refine feature representations using contrastive training. It employs a sub-sampled K-Nearest Neighbor (KNN) classifier to address class imbalance and improve anomaly detection. FS3 was evaluated on WUSTL-EHMS, WUSTL-IIoT, and BoT-IoT datasets, achieving F1-scores above 90%, even with only 20% labeled data. However, the contrastive training step requires careful selection of positive and negative pairs, and the KNN classifier may struggle to scale with large datasets. Expanding on SSL's applicability to structured network data, Nguyen and Kashef [14] proposed TS-IDS, an SSL-based Graph Neural Network (GNN) model that represents IoT network traffic as a graph to capture complex structural relationships. The model refines its graph embeddings through self-supervised pretext tasks and demonstrated superior performance on NF-ToN-IoT and NF-BoT-IoT datasets, outperforming existing GNN-based intrusion detection models. However, the high memory requirements of graph processing and the model's lack of interpretability pose challenges for large-scale deployment and real-world usability.

Overall, SSL-based approaches offer powerful alternatives to traditional semi-supervised learning, improving feature extraction, adaptability in low-data settings, and representation learning for structured data.

C. Ensemble Semi-Supervised Methods for Improved Detection

While both semi-supervised and self-supervised learning methods have their strengths, they also have limitations that can be addressed by having an ensemble approach. Ensemble methods leverage the strengths of multiple models to create a more robust and accurate system.

For instance, Zheng et al [15] proposed an anomaly detection algorithm for IoT devices based on semi-supervised ensemble learning. This method addressed data heterogeneity by categorizing IoT devices into periodic, trend, and random types using time series decomposition. It then employed both direct detection methods (e.g., 3-sigma and similarity-based algorithms) for random devices and indirect detection methods (e.g., prediction or reconstruction-based models) for periodic and trend devices. By selecting the optimal sub-models from different algorithm types, they created a heterogeneous ensemble model that achieved an accuracy of 89.25%. However, the reliance on the availability of a substantial amount of labeled data to categorize devices and train sub-models is a significant limitation. Similarly, Liu et al. [16] presented a semi-supervised ensemble algorithm for IoT anomaly detec-

tion by combining Semi-Supervised Extreme Learning Machines (SSELM) with mutual information-based base classifier selection. Their method dynamically selected classifiers for an ensemble based on maximum relevance and minimum redundancy criteria. While this approach achieved high performance on several UCI datasets, it also has drawbacks. The method assumes a fixed ratio of labeled to unlabeled data during training (e.g., 4:1, 3:2), which may not reflect real-world scenarios where labeled data is extremely scarce. Additionally, it relies on graph Laplacian-based semi-supervised learning, which introduces computational overhead and limits scalability for large-scale datasets like TON-IoT.

These works demonstrate the promise of semi-supervised ensemble methods for IoT anomaly detection but also highlight their dependence on a substantial amount of labeled data, computational complexity, and scalability issues. To address these challenges, our approach adopts an ensemble combining self-supervised models for feature extraction and supervised models for classification. Self-supervised learning excels at learning representations directly from raw data without requiring labels, making it highly suitable for the unlabeled portion of IoT datasets. By integrating supervised models into the ensemble, we aim to leverage labeled data to refine detection performance and improve the accuracy of anomaly classification. This combination enables us to fully utilize the strengths of both paradigms, addressing the scarcity of labeled data while improving anomaly detection in complex, large-scale IoT datasets such as TON-IoT.

In summary, our self-supervised ensemble approach addresses the limitations of semi-supervised methods by minimizing their reliance on labeled data while still incorporating supervised models to achieve high accuracy in detecting anomalies.

III. ToN-IoT Dataset for IoT Security

Selecting a representative and robust dataset is essential for evaluating the effectiveness of intrusion detection models, particularly in the context of IoT environments. Traditional datasets, such as NSL-KDD [17] and CICIDS2017 [18], often fall short in capturing the complexity and diversity of modern IoT ecosystems, which involve heterogeneous data sources, interconnected systems, and dynamic attack vectors.

To bridge this gap, the Telemetry over Networks for IoT (ToN-IoT) dataset was introduced by Alasadi et al. [19] to provide a realistic, comprehensive, and multi-source benchmark tailored for IoT security research. The dataset integrates telemetry data, operating system logs, and network traffic generated across IoT, Information technology (IT), and Operational Technology (OT) environments. This broad scope allows cross-layer intrusion detection and mirrors real-world conditions in smart environments. Booij et al. [20] further analyzed the structure and composition of the ToN-IoT dataset, emphasizing its heterogeneity and the pressing need for standardization in IoT intrusion detection research. Their work highlights the dataset's unique capability to represent modern threats by encompassing both benign and malicious behaviors across diverse sources, thereby positioning ToN-IoT as a superior alternative to earlier benchmarks.

Given the realistic design and comprehensive coverage of IoT data types and threats, we selected the ToN-IoT dataset as the foundation for our experiments. It enables us to rigorously assess our network anomaly detection approach within a realistic and diverse IoT context.

A. Dataset Construction and Subsets

Given that the original ToN-IoT dataset comprises approximately 23 million data points, we derive two controlled subsets to facilitate scalable experimentation while preserving essential data characteristics. Each subset contains exactly 1 million data points but is constructed using different sampling strategies to support evaluation under varying conditions.

1) Dataset 1: Proportional Sampling of Original Distribution: Dataset 1 is obtained by randomly sampling the original dataset while preserving its inherent class imbalance. In the complete dataset, normal (benign) instances represent approximately 3.56% (79,638 out of 22,339,021), whereas abnormal (malicious) instances account for about 96.44% (21,542,641 out of 22,339,021). In Subset A, this imbalance is maintained by including roughly 36,000 normal data points and 964,000 abnormal data points. This subset is intended to mimic environments where attack simulations result in a predominance of malicious samples, providing a challenging scenario for detection systems.

2) Dataset 2: Balanced Toward Normal Traffic: Dataset 2 is designed to reflect scenarios where normal traffic far outweighs malicious activity. In this subset, we deliberately enhance the number of normal samples, creating a distribution of approximately 800,000 normal data points (80%) versus 200,000 abnormal data points (20%). This configuration is more representative of many operational networks, where anomalies are relatively rare. The adjusted distribution facilitates the evaluation of false positive sensitivity and the generalization capability of detection algorithms in less adversarial contexts.

These two subsets allow us to evaluate our models under different conditions: one that preserves the original distribution and another that simulates a more realistic scenario. The detailed distribution of the original dataset and these two subsets is presented in Table I. Also, Table II presents the detailed breakdown of attack types for the complete dataset, dataset 1, and dataset 2. This table helps to illustrate how specific attack types (e.g., DDOS, DOS, Injection, etc.) are represented in each data subset.

TABLE I: Label Distribution in the Complete Dataset, Dataset 1, and Dataset 2 (Count/%)

Label	**Complete** (Count/%)	**Dataset 1** (Count/%)	**Dataset2** (Count/%)
0 (Normal)	7.96e5/3.56%	3.60e4/3.56%	7.96e5/79.64%
1 (Abnormal)	2.15e7/96.44%	9.64e5/96.44%	2.04e5/20.36%

TABLE II: Attack Type Distribution in the Complete Dataset, Dataset 1, and Dataset 2

Attack Type	Complete		Dataset 1		Dataset 2	
	Count	Percentage	Count	Percentage	Count	Percentage
Backdoor	508,116	2.27%	22,725	2.27%	4,796	0.48%
DDOS	6,165,008	27.60%	275,796	27.58%	58,266	5.83%
DOS	3,375,328	15.11%	150,942	15.10%	31,902	3.19%
Injection	452,659	2.03%	20,292	2.03%	4,282	0.43%
MITM	1,052	0.005%	48	0.005%	11	0.001%
Password	1,718,568	7.69%	76,938	7.69%	16,237	1.62%
Ransomware	72,805	0.33%	3,290	0.33%	693	0.07%
Scanning	7,140,161	31.96%	319,576	31.96%	67,492	6.75%
XSS	2,108,944	9.44%	94,324	9.43%	19,941	1.99%

B. Data Splitting for Supervised and Self-Supervised Learning

To thoroughly evaluate our ensemble-based anomaly detection approach, we design experiments under both supervised and self-supervised learning settings using Dataset 1 and Dataset 2. As mentioned previously, for each dataset, we designate 20% of the data as labeled and the remaining 80% as unlabeled, with the latter used exclusively for self-supervised learning. The labeled portion is further split into three sets: 70% for training, 15% for validation, and 15% reserved as a hold-out test set. Both the supervised and self-supervised models are validated using the same labeled data (validation set). The validation set plays a critical role, as we use it to evaluate and compare the performance of the supervised models as well as the self-supervised models. Based on the validation results, we select the best-performing supervised model and the best-performing self-supervised model. Finally, these models are combined into an ensemble, and the hold-out test set is used to assess the performance of the ensemble as well as the individual models. This strategy not only facilitates robust model training but also enables a fair comparison of learning approaches.

IV. Methodology

A. Experimental Setup

As detailed in Sections III-A and III-B, each dataset is partitioned by allocating 20% of the data as labeled and 80% as unlabeled. The labeled data is further divided into training, validation, and test sets. This standard partitioning facilitates robust supervised training, effective self-supervised pre-training, and unbiased performance evaluation.

Supervised models are trained using the labeled training set, while self-supervised models exploit the unlabeled data to learn meaningful representations of raw network traffic. The validation set is then used to assess and compare these approaches, allowing the selection of the best-performing supervised model and the most effective self-supervised representation model. These selected models are combined into a final ensemble whose performance is evaluated on the hold-out test set. This experimental framework minimizes information leakage and overfitting while providing a clear, comprehensive comparison between fully supervised learning and scenarios in which limited labeled data is augmented by abundant unlabeled samples.

B. Evaluation Metrics

To evaluate the effectiveness of our anomaly detection framework, we employ four widely used classification metrics: Precision, Recall, F1-score, and ROC-AUC. These metrics provide complementary insights into the model's ability to detect anomaly, especially in the presence of class imbalance, which is a common characteristic of real-world cybersecurity datasets. Notably, one of our datasets is skewed toward anomalous samples, while the other is skewed toward normal traffic, further necessitating the use of multiple metrics for a better evaluation.

Precision measures the proportion of instances classified as anomalies that are truly anomalous. High precision indicates a low false positive rate, which is critical in security settings to avoid alert fatigue.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

Recall, or sensitivity, measures the proportion of actual anomalies that are correctly identified by the model. It reflects the model's ability to detect true threats and minimize false negatives.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

F1-score is the harmonic mean of precision and recall, providing a single metric that balances both false positives and false negatives. It is particularly useful when the dataset is imbalanced and neither metric alone gives a complete picture of performance.

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

ROC-AUC (Receiver Operating Characteristic – Area Under the Curve) evaluates the model's ability to distinguish between classes across all classification thresholds. A higher ROC-AUC indicates better overall ranking of anomaly scores, regardless of threshold. This metric is especially useful when the class distribution is uneven, as it is insensitive to imbalance.

In the context of anomaly detection, where false negatives can represent missed attacks and false positives can overload analysts, using a combination of these metrics enables a comprehensive and fair evaluation. Our metric selection ensures

that both detection capability and operational practicality are taken into account.

C. Feature Engineering

Effective feature engineering is critical for developing robust supervised models. In our preprocessing pipeline, we first removed columns that either leak target information or fail to generalize across different network environments. Specifically, we discarded columns containing source and destination IP addresses and ports. These features are highly specific to individual network instances and are subject to frequent changes, which can lead to overfitting on particular network conditions rather than enabling the model to learn general patterns of normal and anomalous behavior. Similarly, the attack_type, which closely resembles the target label, was removed to avoid bias and leakage.

After eliminating these features, we performed a correlation analysis on the remaining features using Pearson correlation coefficients [21]. The analysis indicated that most features exhibited only weak to moderate correlations with the target labels, with no single feature dominating the prediction task. This result suggested that while no individual feature alone was highly predictive. Consequently, we chose to retain all remaining features to preserve the full spectrum of potentially useful information.

D. Handling Missing Values and Duplicates

Maintaining high data quality is essential for effective model training. In our datasets, we identified several columns with more than 90% missing values; such columns were removed, as their sparse data was unlikely to contribute meaningfully to the anomaly detection task. In addition, duplicate rows were eliminated across both datasets to prevent data leakage and ensure that every training instance added unique value to the learning process.

E. Supervised Learning Component of the Ensemble

Within our ensemble framework, the supervised component leverages the high-quality labeled subset—despite its limited size—to capture explicit patterns of normal and anomalous behavior in IoT network traffic. Although collecting labeled data is both costly and labor-intensive, this valuable portion enables our models to learn reliable discriminative features. Following the data splitting steps detailed in Section III-B, multiple supervised learning algorithms were trained using the allocated training sets. Their performance was rigorously compared on the validation set, which served to identify the optimal model. Ultimately, only the best-performing supervised model was selected for integration into the final ensemble, contributing strong and confident predictions. This targeted approach not only enhances detection performance but also complements the self-supervised component, ensuring that our overall framework is robust even when labeled data is scarce.

1) Learning Algorithms: To capture a broad range of decision boundaries in IoT network anomaly detection, we evaluated a diverse set of supervised learning algorithms, each with distinct modeling characteristics and strengths. First, we employed the Random Forest [22], a tree-based ensemble method renowned for its ability to model complex, non-linear interactions without requiring extensive feature engineering. Its inherent use of bootstrap aggregation and random feature selection makes it robust against overfitting, while the provision of feature importance estimates aids interpretability. Complementing this, we selected Logistic Regression [23] as a representative linear model. Despite its simplicity, Logistic Regression offers strong interpretability and serves as a reliable baseline by effectively modeling linearly separable patterns. Its coefficient estimates allow for straightforward insight into the influence of individual features, which is particularly beneficial when working with structured, tabular data. In addition, we integrated Naive Bayes [24] into our evaluation framework. This probabilistic model, while based on the simplifying assumptions of feature independence and normal distribution of continuous variables, is computationally efficient and often yields surprisingly robust performance, especially when the training dataset is limited. Its simplicity provides a useful contrast to more complex models and helps establish baseline performance metrics. Finally, we employed a Multilayer Perceptron (MLP) [25] to account for the possibility of non-linear interactions among features. As a feedforward neural network, the MLP is capable of learning deep, non-linear representations that can capture subtle patterns not accessible to simpler models. This flexibility makes it a valuable component in our ensemble despite the additional challenges it poses in terms of hyperparameter tuning and computational cost.

By incorporating these four distinct algorithms, our framework benefits from a comprehensive evaluation across multiple modeling paradigms, ultimately facilitating the selection of the most effective supervised model for integration into our ensemble-based anomaly detection system.

2) Hyperparameter Tuning and Performance Evaluation: To optimize our supervised models for anomaly detection, we adopted a unified hyperparameter tuning strategy using GridSearchCV [26] combined with Stratified K-Fold Cross-Validation [27]. This process minimizes overfitting by splitting the training set into stratified folds that preserve class distributions—an especially critical factor for imbalanced datasets typical of anomaly detection tasks. We used the F1-score as the scoring method during tuning. Upon identification of the optimal hyperparameters, each model was retrained on the entire training set before final evaluation on the validation set. This systematic approach ensures robust performance and strong generalization on unseen data. The model that achieved the highest F1-score and ROC-AUC on the validation set was selected for integration into the final ensemble.

F. Self-Supervised Learning Component of the Ensemble

Self-supervised learning (SSL) has emerged as a powerful paradigm for representation learning in settings where labeled

data is scarce or expensive to obtain. SSL learns meaningful patterns without relying on any labels. This makes SSL particularly well-suited for anomaly detection in IoT networks, where labeling network traffic at scale is often infeasible due to the diversity and volume of data, as well as the evolving nature of cyber threats.

In our framework, SSL is used to extract high-quality feature embeddings from unlabeled IoT network traffic. These embeddings serve as the basis for downstream classification and are integrated into our ensemble system to enhance robustness and generalization.

1) Learning Algorithms: To evaluate the strengths of different self-supervised learning strategies, we adopt two complementary approaches: Variational Autoencoders (VAE) [28] and Contrastive Learning (CL) [29].

Variational Autoencoder (VAE): VAEs are generative models that learn to encode input data into a latent distribution, from which samples can be drawn and decoded back into the input space. This reconstruction mechanism enables the model to capture the underlying distribution of normal and anomalous traffic. We use the mean squared error (MSE) between the original and reconstructed inputs as the anomaly score. For Dataset 1, where anomalous traffic is the majority class, the VAE is trained using the entire unlabeled set, which primarily consists of anomalous samples. The model thus learns a distribution skewed toward anomalies, and deviations from this distribution indicate normal behavior. For Dataset 2, where normal traffic is the majority, the VAE similarly learns the latent representation of normal behavior. In this case, reconstruction errors highlight deviations that are likely to be malicious, aligning with typical anomaly detection use cases. This dual use of VAE across contrasting class distributions allows the model to adaptively define "normal" or "anomaly" based on the data composition, making it a flexible and context-aware method.

Contrastive Learning (CL): Contrastive learning aims to learn embeddings by pulling similar data points closer in the latent space while pushing dissimilar ones apart. We adopt a SimCLR [] architecture that processes positive and negative pairs generated from augmentations of the same or different input samples. The contrastive loss optimizes the model to encode samples with similar semantic meaning close together. After learning the embeddings, we employ a logistic regression classifier trained on the labeled training set to perform binary classification. Logistic regression is chosen due to its simplicity and strong generalization properties, especially when applied to high-quality embeddings. This lightweight classifier also reduces computational overhead, which is crucial for deployment in resource-constrained IoT environments.

2) Hyperparameter Tuning and Model Selection: We perform hyperparameter tuning for both the VAE and contrastive learning approach using the labeled training set. Key parameters such as embedding size, batch size, learning rate, and temperature scaling are optimized to maximize validation performance. For the VAE, tuning includes the latent dimension, learning rate, and reconstruction weight.

To determine the more effective self-supervised model for each dataset, we evaluate both VAE and CL using the labeled validation set. The model that achieves the higher F1-score, and ROC-AUC on the validation data is selected for integration into the ensemble. This selection process ensures that the chosen self-supervised component contributes meaningfully to the final anomaly detection performance.

G. Ensemble and Voting Method

To make detection more reliable and take advantage of the strengths of different learning methods, we built an ensemble framework that combines supervised and self-supervised models. The rationale behind this approach is twofold. First, supervised models (such as Random Forest in our case) have demonstrated strong performance when substantial labeled data is available, while self-supervised models (such as the Contrastive Learning approach) excel at capturing intrinsic data structures even when labels are limited. Combining these models allows the ensemble to benefit from high discriminative power as well as robust feature learning, thereby improving overall anomaly detection performance.

In an ensemble consisting of only two models, a hard voting scheme where the final decision would rely solely on a majority vote can lead to ambiguity when the two models disagree. Since there is no natural majority vote in a two-model scenario, we opt for a *soft voting* strategy. Soft voting combines the prediction probabilities (confidence scores) from the two models rather than their binary outputs. This method provides a more nuanced decision by accounting for each model's confidence in its prediction. Specifically, the final ensemble probability, P_{ens}, is calculated as a weighted average:

$$P_{ens} = \alpha \times P_{ssl} + (1 - \alpha) \times P_{sup}, \tag{4}$$

where P_{sup} is the prediction probability from the supervised model, P_{ssl} is from the self-supervised model, and $\alpha \in [0, 1]$ is a tunable parameter that balances their relative influences. A decision threshold is subsequently applied to P_{ens} to yield the final classification.

Ensemble Tuning: To identify the optimal weighting factor (α) and decision threshold, we conducted a grid search on the labeled validation set with the goal of maximizing the F1-score. Figure 1 illustrates the grid search process.

Specifically, we explored the following parameter ranges:

- Weighting Factor (α): $\{0.0, 0.2, 0.4, 0.6, 0.8\}$.
- Decision Threshold: $\{0.1, 0.2, \ldots, 0.9, 1.0\}$.

The combination of parameters resulting in the highest F1-score on the validation set was selected for the final ensemble evaluation on the test set. This methodical tuning ensures that the ensemble is optimally configured to harness the strengths of both the supervised and self-supervised components, ultimately leading to improved anomaly detection performance.

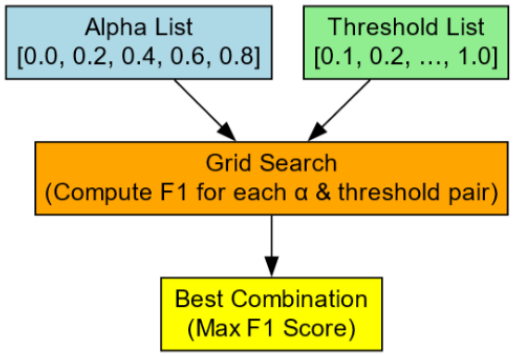


Fig. 1: Grid search methodology for tuning the ensemble weighting factor (α) and decision threshold to maximize the F1-Score on the validation set.

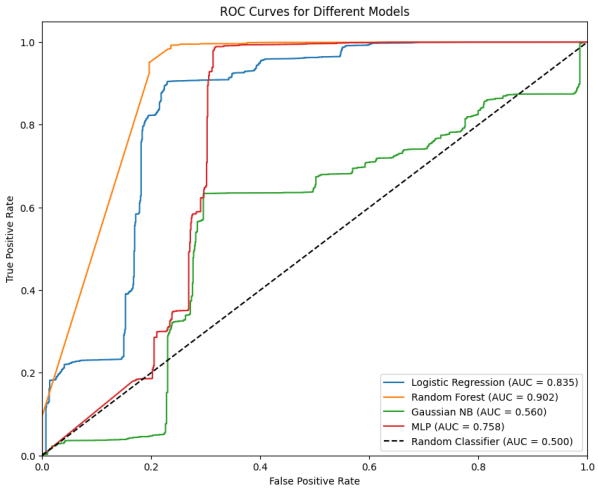


Fig. 2: ROC curves for supervised models on Dataset 1 (Validation Set)

V. Experimental Results and Discussion

A. Supervised Learning Results

Table III presents the F1-scores and ROC-AUC values for each supervised model evaluated on two distinct validation sets. Notably, the models' performance varies considerably between Dataset 1 (anomaly-majority) and Dataset 2 (normal-majority), indicating the influence of dataset characteristics on model behavior.

For Dataset 1, all models except Naive Bayes achieved exceptionally high F1-scores (≈ 0.987–0.989). This outcome suggests that when anomalies are prevalent, the supervised classifiers are highly effective in correctly classifying the minority class. However, a closer look at the ROC-AUC values reveals nuanced differences in discriminative power. In particular, the Random Forest model, despite attaining an F1-score of approximately 0.9887, yielded an ROC-AUC of 0.9016—indicating strong overall discriminative performance. The MLP produced a slightly higher F1-score (≈ 0.9893) but demonstrated a noticeably lower ROC-AUC of 0.7578. This lower value may suggest that the MLP, despite being able to label observations correctly under the specific threshold used for the F1-score, may suffer from overfitting or offer less robust separation between classes when assessed across all thresholds. Logistic Regression performed consistently with an F1-score of 0.9867 and a moderate ROC-AUC of 0.8354. Naive Bayes significantly underperformed, as evidenced by both its F1-score (≈ 0.0050) and ROC-AUC (≈ 0.5597), indicating an inability to capture the complexity of the underlying data distribution on Dataset 1.

In contrast, for Dataset 2, the overall F1-scores are lower, which is expected given that Dataset 2 is normal-majority and poses a different challenge in terms of class imbalance. The MLP achieved the highest F1-score (≈ 0.8739) alongside an excellent ROC-AUC of 0.9713. This suggests that the MLP is particularly adept at capturing subtle patterns under normal-majority conditions, yielding high sensitivity. The Random Forest model shows robust discriminative capabilities (ROC-AUC ≈ 0.9720) with a slightly lower F1-score (≈ 0.7915). This stable ROC-AUC across both datasets highlights its adaptability to different class distributions, even if its threshold-dependent performance (F1-score) does not always peak. Logistic Regression and Naive Bayes again trail behind the top performers, with Logistic Regression recording a F1-score of 0.6948 and ROC-AUC of 0.9043, and Gaussian Naive Bayes showing particularly limited effectiveness (F1-score ≈ 0.3818; ROC-AUC ≈ 0.6954).

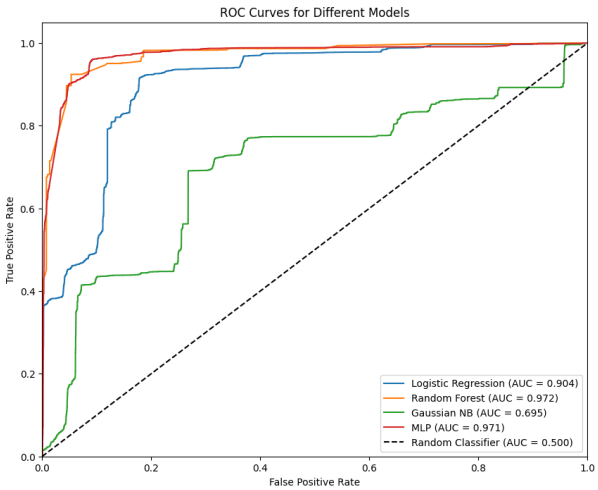


Fig. 3: ROC curves for supervised models on Dataset 2 (Validation Set)

In addition to the tabulated metrics, the ROC-AUC curves depicted in Figures 2 and 3 visually illustrate the discriminative performance of each model across the two datasets. These figures underscore the differences observed in the tabulated ROC-AUC values, further reinforcing that while the MLP exhibits strong threshold-based performance, its overall class

TABLE III: Supervised Model Evaluation Metrics Across Datasets (on Validation Sets)

	Dataset 1		Dataset 2	
Model	**F1-score**	**ROC-AUC**	**F1-score**	**ROC-AUC**
Random Forest	0.9887	0.9016	0.7915	0.9720
Logistic Regression	0.9867	0.8354	0.6948	0.9043
MLP	0.9893	0.7578	0.8739	0.9713
Gaussian Naive Bayes	0.0050	0.5597	0.3818	0.6954

separation in Dataset 1 is less robust compared to Random Forest.

The contrasting performance metrics across the two datasets imply that while the MLP has strong threshold-dependent performance (F1) in both scenarios, its lower ROC-AUC in Dataset 1 raises concerns about its generalization ability in a setting with a high anomaly rate. On the other hand, the Random Forest model not only provides a balanced performance between the F1-score and ROC-AUC metrics but also demonstrates consistent discriminative strength across both types of datasets. Consequently, the Random Forest model emerges as a strong candidate for integration into the final anomaly detection ensemble.

B. Self-Supervised Learning Results

1) Variational Autoencoder Results: The Variational Autoencoder was evaluated on both datasets to assess its ability to model network behavior in an unsupervised setting using reconstruction error. Table IV summarizes the classification performance, and Figure 4 illustrates the ROC curves.

TABLE IV: VAE Validation Performance Across Datasets

Dataset	**Precision**	**Recall**	**F1-score**
Dataset 1 (Anomaly-Majority)	0.64	0.26	0.36
Dataset 2 (Normal-Majority)	0.21	0.50	0.30

While VAE achieved moderate AUCs on both datasets, its overall classification performance was weak—particularly in terms of recall. On Dataset 1, the model exhibited poor generalization to normal samples, likely because it was exposed primarily to anomalies during training. As a result, the model reconstructed anomalies too well, reducing reconstruction error and misclassifying them as normal.

On Dataset 2, where normal traffic was dominant, the model still failed to capture the distinguishing characteristics of rare attacks. This reinforces a key limitation of VAEs: they are sensitive to class distribution and assume the majority class defines "normal." When this assumption is violated (as in Dataset 1), or when anomalies are subtle (as in Dataset 2), VAE struggles to separate benign from malicious behavior effectively.

2) Contrastive Learning Results: Contrastive Learning was evaluated on the same datasets, using embeddings learned via a Siamese architecture and a logistic regression classifier for downstream anomaly detection. Table V shows the evaluation metrics, and Figure 5 presents the ROC curves.

TABLE V: Contrastive Learning Validation Performance

Dataset	**Precision**	**Recall**	**F1-score**
Dataset 1 (Anomaly-Majority)	0.89	0.89	0.89
Dataset 2 (Normal-Majority)	0.99	1.00	0.99

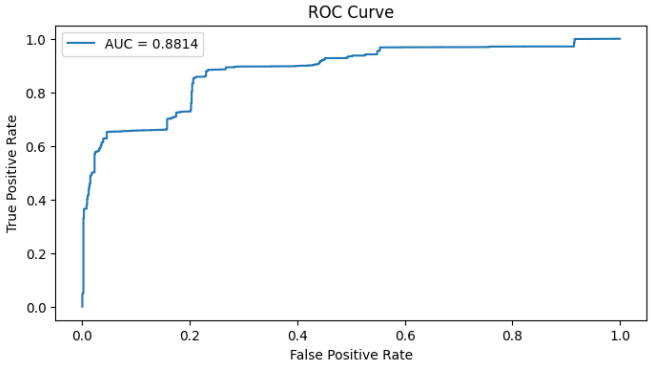
CL demonstrates strong and consistent performance across both datasets. On Dataset 1, the method achieved a balanced F1-score of 0.89 despite the majority of samples being anomalous, suggesting excellent representation learning even under skewed distributions. On Dataset 2, where the data is more balanced, CL yielded nearly perfect performance, with a 0.99 F1-score and near total recall. This resilience stems from CL's core mechanism: learning to distinguish semantically different samples through contrastive objectives. Unlike VAE, which learns to reconstruct inputs, CL focuses on relational understanding in the feature space. This enables it to identify anomalies not by how they differ from the majority class in raw form, but by how they deviate in learned semantics even if they are subtle or rare.

3) Method Comparison and Selection: The results indicate that Contrastive Learning significantly outperforms Variational Autoencoders in the context of IoT anomaly detection. VAE is particularly vulnerable to data imbalance and fails to generalize when the definition of "normal" is fluid or corrupted by anomalies. In contrast, CL excels at capturing abstract similarities and differences that are robust to distributional skew and better suited for real-world, noisy, and evolving network traffic.

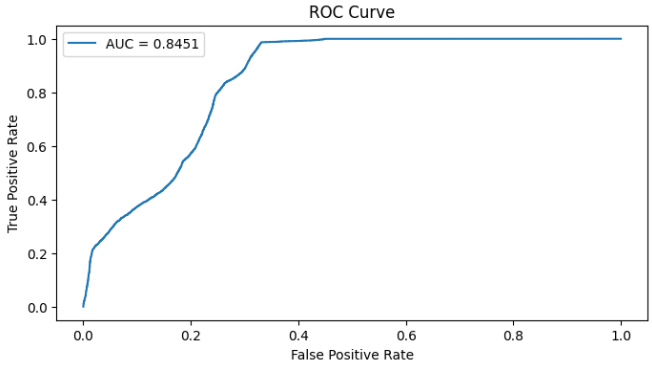
Given its superior performance across all key metrics, precision, recall, F1-score, and AUC—Contrastive Learning is selected as the preferred self-supervised model in our ensemble. It offers high detection accuracy, better generalization, and efficient integration with downstream classifiers, making it a reliable and scalable choice for detecting anomalies in IoT networks.

C. Final Ensemble Results and Comparative Evaluation

The final ensemble, optimized via the grid search procedure described in Section IV-G, was evaluated on the held-out test sets for both Dataset 1 (anomaly-majority) and Dataset 2 (normal-majority). Figure 6 presents a visual comparison of the F1-Score and ROC-AUC metrics achieved by the individual supervised (Random Forest) and self-supervised (Contrastive Learning) models alongside those of the tuned

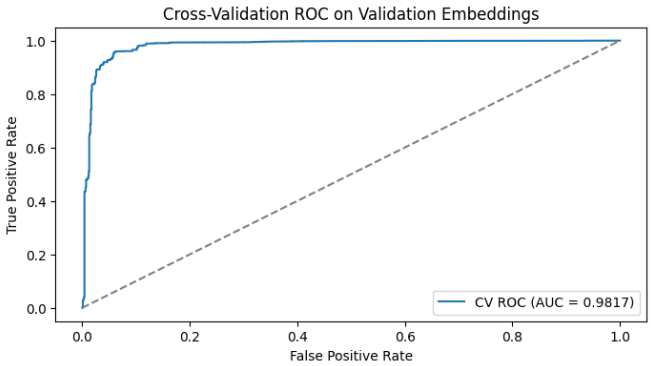


(a) ROC Curve – Dataset 1 (AUC = 0.8451)

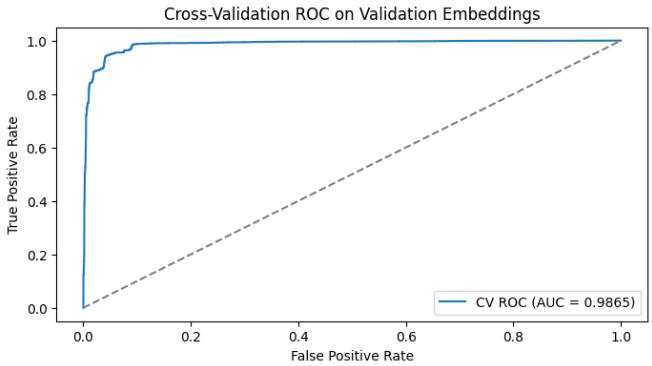


(b) ROC Curve – Dataset 2 (AUC = 0.8814)

Fig. 4: VAE ROC-AUC Curves on Two Datasets (Validation Sets)



(a) ROC Curve – Dataset 1 (AUC = 0.9865)



(b) ROC Curve – Dataset 2 (AUC = 0.9817)

Fig. 5: Contrastive Learning ROC-AUC Curves on Two Datasets (Validation Sets)

ensemble. Table VI details the key performance metrics (Precision, Recall, F1-score, and ROC-AUC) for all three approaches across the two datasets.

A closer analysis of the results reveals several key observations:

Analytical Discussion

Dataset 1 (Anomaly-Majority): In Dataset 1, where anomalies are common, all models achieve high performance. Nevertheless, the ensemble model clearly improves upon the individual approaches, reaching an F1-Score of 0.9937 and an ROC-AUC of 0.9862. Figure 6 corroborates these improvements by showing that the ensemble not only outperforms the supervised approach but also leverages the high confidence of the self-supervised model. The enhanced F1-score indicates that the ensemble successfully reduces false positives while ensuring that nearly all anomalous instances are detected.

Dataset 2 (Normal-Majority): For Dataset 2, characterized by a significant class imbalance, the self-supervised model (Contrastive Learning) achieves the highest F1-Score (0.905) among the individual models, while the ensemble model records a very similar F1-Score of 0.9045. Although the ensemble does not markedly exceed the self-supervised approach in terms of F1-Score, it does achieve a slightly higher ROC-AUC (0.9860 vs. 0.985). This suggests that, even when the self-supervised method is dominant, the ensemble method preserves competitive performance while providing additional robustness, particularly regarding overall discriminative capacity.

Overall Insights: The ensemble approach consistently delivers robust performance across both datasets. On Dataset 1, it clearly surpasses the individual models, while on Dataset 2, it matches the best performing approach with marginal gains in overall discrimination (as measured by ROC-AUC). These outcomes demonstrate that combining predictions via soft voting—wherein each model's confidence is utilized—provides a more resilient and balanced anomaly detection system compared to relying on either approach in isolation.

D. Limitations

While our ensemble framework demonstrates robust performance for IoT network anomaly detection, several limitations remain that highlight important trade-offs and areas for improvement:

- Performance vs. Complexity Trade-off: Our approach integrates a deep self-supervised model based on Contrastive Learning with a traditional supervised model (Random Forest). Although the ensemble achieves high

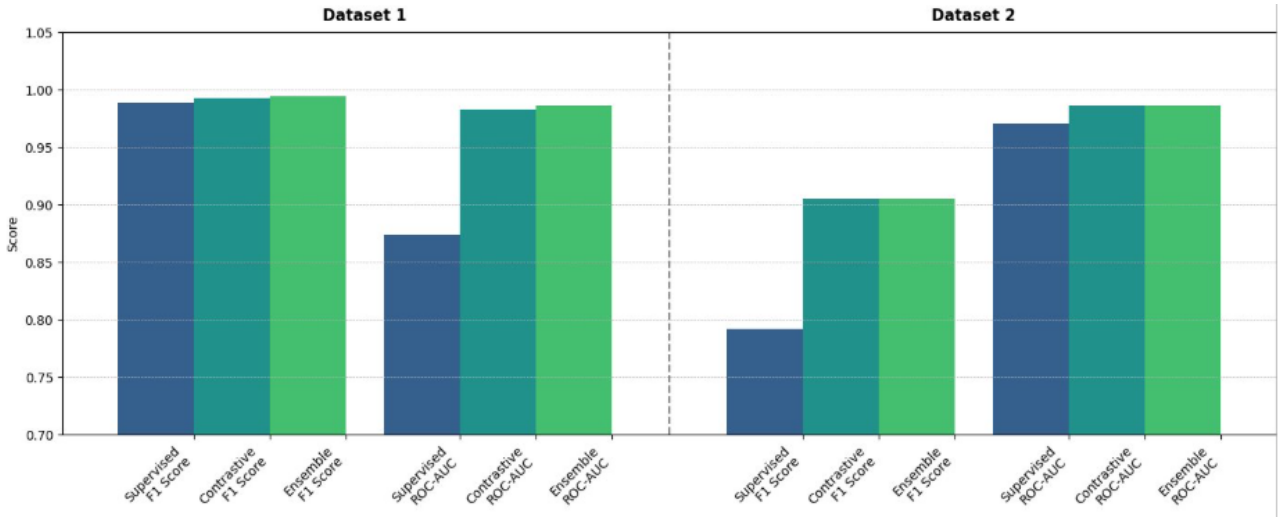


Fig. 6: Comparison of F1-Score and ROC-AUC metrics for the best Supervised (RF), Self-Supervised (CL), and Ensemble models on the test sets for Dataset 1 and Dataset 2.

TABLE VI: Comparison of Supervised, Self-Supervised, and Ensemble Models Across Datasets (Test Sets)

	Dataset 1 (Anomaly-Majority)			Dataset 2 (Normal-Majority)		
Metric	**Supervised (RF)**	**Self-Supervised (CL)**	**Ensemble**	**Supervised (RF)**	**Self-Supervised (CL)**	**Ensemble**
Precision	0.990	0.991	**0.9941**	0.790	**0.905**	0.8659
Recall	0.875	0.982	**0.9933**	0.970	**0.985**	0.9468
F1-Score	0.990	0.991	**0.9937**	0.790	**0.905**	0.9045
ROC-AUC	0.875	0.982	**0.9862**	0.970	0.985	**0.9860**

Notes: The table reports Precision, Recall, F1-Score, and ROC-AUC for Dataset 1 (anomaly-majority) and Dataset 2 (normal-majority). **Bold** values indicate the best performance among the models for each metric. Abbreviations: RF = Random Forest, CL = Contrastive Learning.

performance, the deep learning component incurs substantial computational overhead and complexity. This trade-off limits its feasibility for real-time or resource-constrained IoT deployments. In practice, the gains in detection performance must be weighed against the increased training time, memory usage, and energy consumption of deep models.

- Limited Model Diversity: The ensemble currently comprises only two models. While this setup leverages the complementary strengths of supervised and self-supervised learning, the restricted diversity may limit the system's adaptability to a wider range of network behaviors. Incorporating additional models could potentially capture more diverse patterns and further enhance detection robustness.
- Sensitivity to Data Distribution and Concept Drift: Our evaluations were conducted on controlled subsets of the ToN-IoT dataset. IoT network traffic, however, is highly dynamic, and shifts in attack patterns (concept drift) can degrade model performance over time. Although our ensemble helps mitigate the impact of imbalanced data, a static model may still struggle to adapt to evolving threats. Online learning strategies are helpful for maintaining performance in operational environments.
- Ensemble Integration Challenges: With only two models, hard voting becomes ambiguous, necessitating our use of soft voting. While soft voting effectively utilizes each model's prediction confidence, it also reveals a limitation: in scenarios where one model consistently outperforms the other (as observed in Dataset 2), the benefit of combining the two can be marginal. More dynamic or multi-model ensemble strategies could potentially offer a more significant performance boost.

VI. Conclusion

In this work, we introduced a semi-supervised ensemble framework for IoT network anomaly detection that effectively combines the strengths of supervised and self-supervised learning. By fusing the predictions of a Random Forest classifier with those from a Contrastive Learning model through a soft voting mechanism, our approach significantly reduces reliance on extensively labeled data while achieving high detection performance. Our evaluation of the ToN-IoT dataset demonstrates that the ensemble model attains high F1-Scores and ROC-AUC values in both anomaly-majority and normal-majority scenarios. In environments with abundant anomalies, the ensemble not only enhances overall performance by balancing precision and recall but also improves the discriminative ability of the detection system. In more imbalanced, normal-majority settings, the ensemble maintains competitive

performance, offering marginal gains in ROC-AUC compared to the self-supervised approach alone.

Looking ahead, future research can extend and refine our approach in several exciting ways. One promising direction is to explore a wider variety of model architectures to enrich the ensemble's representational capacity. Incorporating additional model families—such as lightweight deep neural networks—could further enhance detection performance by capturing a broader spectrum of network behavior. Advancing adaptive or online learning techniques to continuously update model parameters in response to the dynamic nature of IoT traffic will also be critical, ensuring that the framework remains responsive to evolving attack patterns. Additionally, efforts to optimize the computational efficiency of deep self-supervised models will help facilitate real-time deployment on resource-constrained IoT devices. Finally, validating the proposed framework on additional large-scale and heterogeneous datasets will provide further insights into its scalability and practical application in varied real-world environments.

Overall, our self-supervised ensemble framework provides a strong foundation for next-generation IoT anomaly detection systems, and the envisioned research directions promise to further advance its efficacy and adaptability in a rapidly evolving cybersecurity landscape.

REFERENCES

[1] A. Dainotti, A. Pescape, and K. C. Claffy, "Issues and future directions in traffic classification," *IEEE network*, vol. 26, no. 1, pp. 35–40, 2012.

[2] S. Rathore and J. H. Park, "Semi-supervised learning based distributed attack detection framework for iot," *Applied Soft Computing*, vol. 72, pp. 79–89, 2018.

[3] A. Sánchez-Ferrera, B. Calvo, and J. A. Lozano, "A review on self-supervised learning for time series anomaly detection: Recent advances and open challenges," *arXiv preprint arXiv:2501.15196*, 2025.

[4] N. Ravi and S. M. Shalinie, "Semisupervised-learning-based security to detect and mitigate intrusions in iot network," *IEEE Internet of Things Journal*, vol. 7, no. 11, pp. 11 041–11 052, 2020.

[5] M. Abdel-Basset, H. Hawash, R. K. Chakrabortty, and M. J. Ryan, "Semi-supervised spatiotemporal deep learning for intrusions detection in iot networks," *IEEE Internet of Things Journal*, vol. 8, no. 15, pp. 12 251–12 265, 2021.

[6] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *Proceedings of the 20th International conference on Machine learning (ICML-03)*, 2003, pp. 912–919.

[7] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the eleventh annual conference on Computational learning theory*, 1998, pp. 92–100.

[8] I. Triguero, S. García, and F. Herrera, "Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study," *Knowledge and Information systems*, vol. 42, pp. 245–284, 2015.

[9] W. Li, W. Meng, and M. H. Au, "Enhancing collaborative intrusion detection via disagreement-based semi-supervised learning in iot environments," *Journal of Network and Computer Applications*, vol. 161, p. 102631, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804520301053

[10] W. Song, X. Li, P. Chen, J. Chen, J. Ren, and Y. Xia, "A novel graph structure learning based semi-supervised framework for anomaly identification in fluctuating iot environment," *CMES - Computer Modeling in Engineering and Sciences*, vol. 140, no. 3, pp. 3001–3016, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1526149224000432

[11] L. Ruff, R. A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K.-R. Müller, and M. Kloft, "Deep semi-supervised anomaly detection," in *International Conference on Learning Representations (ICLR)*, 2020.

[12] S. Aktar and A. Y. Nur, "Robust anomaly detection in iot networks using deep svdd and contractive autoencoder," in *2024 IEEE International Systems Conference (SysCon)*. IEEE, 2024, pp. 1–8.

[13] D. Ayesha S and S. AB, "Fs3: Few-shot and self-supervised framework for efficient intrusion detection in internet of things networks," in *Proceedings of the 39th Annual Computer Security Applications Conference*, 2023, pp. 138–149.

[14] H. Nguyen and R. Kashef, "Ts-ids: Traffic-aware self-supervised learning for iot network intrusion detection," *Knowledge-Based Systems*, vol. 279, p. 110966, 2023.

[15] J. Zheng, Y. Xiang, and S. Li, "Anomaly detection algorithm of iot devices based on semi-supervised heterogeneous ensemble learning," in *2022 2nd International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology (CEI)*. IEEE, 2022, pp. 807–813.

[16] S. Liu, X. Hao, and X. Chen, "A semi-supervised dynamic ensemble algorithm for iot anomaly detection," in *2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)*. IEEE, 2020, pp. 264–269.

[17] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *2009 IEEE symposium on computational intelligence for security and defense applications*. Ieee, 2009, pp. 1–6.

[18] I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani *et al.*, "Toward generating a new intrusion detection dataset and intrusion traffic characterization." *ICISSp*, vol. 1, no. 2018, pp. 108–116, 2018.

[19] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "Ton_iot telemetry dataset: A new generation dataset of iot and iiot for data-driven intrusion detection systems," *Ieee Access*, vol. 8, pp. 165 130–165 150, 2020.

[20] T. M. Booij, I. Chiscop, E. Meeuwissen, N. Moustafa, and F. T. H. den Hartog, "Ton_iot: The role of heterogeneity and the need for standardization of features and attack types in iot network intrusion datasets," *IEEE Internet of Things Journal*, 2021.

[21] J. Benesty, J. Chen, and Y. Huang, "On the importance of the pearson correlation coefficient in noise reduction," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 4, pp. 757–765, 2008.

[22] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. [Online]. Available: https://link.springer.com/article/10.1023/A:1010933404324

[23] P. McCullagh and J. A. Nelder, *Generalized Linear Models*, 2nd ed., ser. Monographs on Statistics and Applied Probability. Chapman and Hall, 1989, vol. 37.

[24] D. D. Lewis, "Naive (bayes) at forty: The independence assumption in information retrieval," in *European conference on machine learning*. Springer, 1998, pp. 4–15.

[25] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986. [Online]. Available: https://www.nature.com/articles/323533a0

[26] S. learn developers, *GridSearchCV: Exhaustive search over specified parameter values for an estimator*, accessed: 2023-06-01. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

[27] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. Springer, 2009. [Online]. Available: https://link.springer.com/book/10.1007/978-0-387-84858-7

[28] D. P. Kingma, M. Welling *et al.*, "Auto-encoding variational bayes," 2013.

[29] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PmLR, 2020, pp. 1597–1607.