International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18 (2) 2026 pp. 11-25

https://doi.org/10.70917/ijcisim-2026-2145
Article

HAEnRF: A Hybrid Deep Autoencoderensemble Framework For Cloud-Agnostic Intrusion Detection

Vishnu Priya P. M.[1], Soumya S.[2]

[1]Institute of Computer Science and Information Science, Srinivas University, Mangalore, Karnataka, India.
vpriyapm@gmail.com
ORCID: 0009-0002-0229-2759

[2] Institute of Computer and Information Sciences, Srinivas University, Mangalore, Karnataka, India.
ORCID: 0000-0002-5431-1977

Abstract: Cloud computing environments present a huge amount of diverse network traffic. This poses a major hurdle in identifying malicious activities using traditional signature-based intrusion detection systems. To tackle this problem, the paper introduces HAEnRF, a hybrid intrusion detection system that integrates multiple autoencoder-based feature learning methods with a supervised Random Forest classifier for detecting flow-level cyberattacks in cloud environments.The proposed system is built and tested with the CICIDS2017 dataset, and a well-defined data preprocessing workflow is set up to ensure numerical stability, balanced class distribution, and consistency during the whole learning process. To capture different traffic characteristics, three types of autoencoder architectures are used: a dense autoencoder, a denoising autoencoder, and a 1D convolutional autoencoder. These models not only capture compact latent feature representations but also output reconstruction errors that can serve as indicators of traffic pattern changes.The latent features and reconstruction errors derived from the autoencoders are merged at the feature level to create a richer feature space. This combined representation is then fed to a Random Forest classifier that has been set up with a balanced class weighting scheme to enhance classification accuracy across various types of attacks. By combining unsupervised representation learning with supervised classification based on ensemble methods, the proposed HAEnRF framework seeks to deliver a more effective and flexible solution for intrusion detection in ever-changing cloud environments.The design of the architecture is both cloud-agnostic and attack-agnostic. This means that the framework can be retrained on other flow-based datasets without making any changes to the main model structure. The present work is mainly concerned with the methodological/architectural design, and conceptualization of the HAEnRF framework. A feasibility-level preliminary evaluation is performed using the CICIDS2017 dataset to prove that the approach is viable. Considering broader benchmarking and cross-dataset validation, these are regarded as future research directions.
Keywords: Network Security, Autoencoder, Random Forest, Anomaly Detection, CICIDS2017, Cloud Computing.

1.INTRODUCTION

Cloud computing has become a major platform for the deployment of various applications, ranging from enterprise services to large-scale data processing systems, as well as cloud-native platforms. This shift undoubtedly offers many benefits such as scalability, flexibility, and better utilization of resources. However, it also opens up several security concerns. As more critical services and sensitive data are hosted on cloud infrastructures, there is a larger potential for security breaches. As a result, cloud networks face a number of vulnerabilities including distributed denial-of-service (DDoS) attacks, botnet activities, and data exfiltration that can lead to both lack of service and data compromise [4], [12]. Besides, attacks in these scenarios might come not only from outside the cloud environment but also from the inside through compromised nodes. So, depending on perimeter defense alone is unlikely to provide the necessary level of security for modern cloud architectures [18].

In spite of the fact that Intrusion Detection System (IDS) technologies are still very important for protecting networks by keeping an eye on the traffic and identifying behaviors that are suspicious or unauthorized, their deployment inside cloud environments is a different story with challenges. To begin with, the network traffic in cloud environments is very heterogeneous because it represents multiple tenants, various applications, and changing communication patterns. On top of that, attributes of the traffic can be altered very quickly with cloud's elastic resource scaling, automated orchestration, and frequent application of software

updates. In the meanwhile, attackers are going to great lengths to use adaptive attack techniques such as zero-day exploits and stealthy multi-stage campaigns which are very difficult to detect through static signatures or predefined rule sets [4], [12].

In order to overcome these issues, Machine Learning and Deep Learning based methods have been broadly acknowledged as an excellent tool that is being utilized not only nowadays but also likely in the future for building modern intrusion detection systems. Algorithms based on learning are capable of automatically identifying patterns from network traffic data, and report anomalies and relationships in a very subtle manner when compared to the traditional rule-based systems that are only able to detect changes at a more obvious and explicit level. Supervised learning methods have been able to show a detection performance that is almost similar to human level when they have access to adequately labeled datasets, while unsupervised methods are very efficient in modeling normal traffic behavior and identifying deviations from it [11], [12].

Among the different unsupervised learning methodologies, autoencoders are one of most popular techniques since they are able to write down very accurate descriptions of the normal network traffic by learning the compact feature representations. These kinds of neural networks take the input data, think about how to rebuild that data from a compressed form, and then use the difference between original and rebuilt data to distinguish normal from anomalous patterns. Past work indicates that autoencoder-based models are not only capable of mirroring the distribution of typical traffic but also detecting abnormal flows even without having the knowledge of attack signatures [5], [6]. Besides that, review studies have revealed that learning-based IDS strategies quite frequently show better adaptability when confronted with high-dimensional network traffic datasets [4], [12].

For instance, although the advantages are considerable, many existing research based on this primarily focus on performance of classification on specific dataset or limited categories of attack. This kind of approach may not be generalised to work well in other cloud environments or changing threat landscapes. To overcome such a weakness, some recent works have proposed hybrid intrusion detection systems that combine different learning methods. A basic theory is that various models can reflect different facets of network behavior. Unsupervised models, e.g. autoencoders are good at learning data representation and detecting anomalies. On the other hand, supervised classifiers can utilize labeled data for drawing precise decision boundaries [7], [8].Having these points in mind, the paper puts forward HAEnRF, a hybrid intrusion detection system combining various autoencoder-based representation learning modules with a Random Forest classifier using an ensemble. The main concept behind the idea is to obtain different representations of network traffic by the means of unsupervised autoencoders and later integrate their results in one supervised classification stage. Instead of depending on just one autoencoder design, HAEnRF uses three different and complimentary versions of autoencoder: a dense (fully connected) autoencoder, a denoising autoencoder, and a one-dimensional convolutional autoencoder. With the use of these three architectural forms, the framework is expected to be able to describe various traffic situations and thereby improve the reliability of intrusion detection under the conditions of rapidly changing cloud environments. The primary contributions of this wok include,

- A hybrid intrusion detection framework (HAEnRF) is proposed that integrates multiple autoencoder-based unsupervised representation learners with a supervised Random Forest classifier, enabling feature-level fusion of anomaly-sensitive and discriminative information within a unified architecture.

- A structured data preprocessing and preparation workflow is defined for flow-level intrusion detection, including feature cleaning, label encoding, class imbalance handling through oversampling techniques, and feature normalization, to support stable and unbiased model training.

- The design rationale for employing dense, denoising, and one-dimensional convolutional autoencoders is explicitly articulated, highlighting how different architectural inductive biases can capture complementary characteristics of network traffic behavior.

- A feature fusion strategy combining latent embeddings and reconstruction error signals from multiple autoencoders is formulated, providing a compact and informative representation suitable for ensemble-based supervised classification.

- A standardized experimental design and evaluation protocol is specified, including dataset partitioning and commonly adopted intrusion detection metrics, to support future empirical benchmarking and reproducible validation of the proposed framework.

- A cloud-agnostic and attack-agnostic architectural design is presented, allowing the framework to be retrained on different labeled flow-based datasets without modification to the core detection pipeline.

The HAEnRF framework differs from most of the hybrid intrusion detection systems that first integrate

autoencoders and classifiers in sequential pipelines or just combine models at the decision level. HAEnRF feature-level fusion strategy that simultaneously incorporates latent representations and reconstruction error signals from multiple heterogeneous autoencoder architectures without feature-level fusion of latent representations and error signals. It is through the joint exploitation of dense, denoising, and convolutional autoencoders that the single unified Random Forest classifier, thereby a decision tree ensemble, is formed. Such a model structure refrains from using either pre-determined or data-driven anomaly thresholds, which are typically set in unsupervised autoencoder-based detection approaches. Instead, a supervised learning model is capable of discovering (H) patterns in input space that are highly sensitive to anomalies directly from the data.

The paper is organized as follows: Section 2 consists of related works, Section 3 consists of the proposed HAEnRF framework that includes preprocessing of data, autoencoder-based representation learning, and RF classification. Section 4 consists of experimental works. Section 5 presents a design-focused analysis of the framework and its expected behavior in cloud intrusion detection scenarios. Section 6 includes the conclusion and future works.

2. RELATED WORK

Research on intrusion detection systems (IDS) has evolved along multiple directions as network environments have become more complex and data-driven methods have matured. The existing IDS research can be classified into three overlapping groups. They are supervised machine learning–based approaches, unsupervised anomaly detection methods, and hybrid frameworks that combine elements of both. Here the works done in each category is discussed as these approaches directly inform the design of the proposed HAEnRF framework.

Supervised Machine Learning- Based IDS:

Support Vector Machine, Decision Tree classifiers, and ensemble methods are some of the widely explored supervised models for intrusion detection that aim to find discriminative decision boundaries to separate benign traffic from malicious activity [4], [12]. Out of these methods, the Random Forest classifier stands out as a popular choice because it can perform well in high-dimensional feature spaces, reduce overfitting through ensemble aggregation, and remain robust even when handling imbalanced datasets [3]. Numerous studies have highlighted the excellent detection performance of Random Forest models when applied to benchmark intrusion detection datasets. For example, after suitable preprocessing and feature engineering, Chavan and Alone attained high classification accuracy on the CICIDS2017 dataset [11]. Ensemble variants combining multiple tree-based learners have also been the subject of other studies, which often show better results compared with single-model baselines [12]. These findings support the idea that Random Forest–based models are very good for flow-level intrusion detection situations where traffic features are structured, heterogeneous, and potentially include redundant information.Besides traditional machine learning techniques, supervised deep learning models have also been used to detect intrusions. For example, James Shone et al. presented a deep learning system based on stacked Restricted Boltzmann Machine layers, which achieved good results on standard intrusion detection datasets [14]. Also, Chuanlong Yin et al. explored the adoption of Recurrent Neural Network models to detect temporal dependencies in network traffic and proved that these architectures can identify sequential attack patterns efficiently [15]. Other researches have been concentrated on developing lightweight deep learning models to optimize the trade-off between computational efficiency and detection performance, hence making them more feasible for real deployment in resource-limited environments [16], [17].Even though they are highly capable of predicting, supervised methods usually require a very large set of data with very accurately labeled instances. This need for predefined attack labels may hinder the ability of such systems to recognize new threats or attack tactics that are changing in real time, thus stressing the importance of looking at other methods that could provide better adaptability in dynamic network environments.

Unsupervised Autoencoder-Based IDS:

Unsupervised methods of intrusion detection aim at identifying unusual behaviors without the need for labeled data about attacks. This makes them very attractive especially in changing or poorly-labeled environments. Autoencoders are among the top choices of models used in this category. When they are mostly trained on the normal benign traffic, autoencoders try to reconstruct normal patterns leading to minimum error, on the other hand, anomalous or malicious flows usually result in higher reconstruction errors. It has been shown in the literature that the simplest stacked autoencoders and the more complex variants can both be very effective for intrusion detection. Song et al. provide the most comprehensive details of different autoencoder architectures and mention that reconstruction-based methods can reach the detection capability at a high level in several datasets [5]. To give evidence that mixing autoencoders makes the system resilient to multiple types of dynamics of traffic, Mirsky et al. introduced Kitsune, which is an ensemble-based system where many autoencoders are trained on fractions of features [6].

Recent research has been focusing on convolutional and recurrent autoencoder architectures mainly to their ability to model sequential or spatial patterns in network data [10]. However, even though these are great advantages, IDS systems that rely on autoencoders only, face several difficulties in real-life situations. To identify normal and abnormal traffic, a number of works depend on fixed thresholds that are applied to reconstruction error only. If the model is not able to sufficiently represent the variety of standard behavior, then setting the right threshold might lead to very high false positive rates, which in addition might be specific to the dataset only. Apart from that, instead of uniting several learned representations directly into a single detection system, many existing works focus on a single autoencoder architecture or even carry out one model at a time.

Hybrid Intrusion Detection Frameworks:

One of the major trends among the studies going forward has been an interest in hybrid intrusion detection frameworks that merge the two paradigms so as to deal with the limitations of simple supervised or unsupervised methods. The basic idea of hybrid methods is to take advantage of unsupervised models at the stage of representation learning or anomaly characterization, while supervised classifiers are employed to make the final detection decision based on the availability of the labeled data. In the two-layer hybrid IDS system proposed by Wang et al., a Random Forest classifier classifies the input data then an autoencoder is used to re-classify doubtful samples [7]. Li et al. managed to achieve not only better detection accuracy but also processing efficiency when they introduced an autoencoder classifier on top of Random Forest-based feature selection [8]. These researches show how the combined usage of tree-based models and neural representations can bring out the performance to a level that is higher than using each of the methods alone.Besides, hybrid ensembles of large-scale approaches have been studied. Almuhanna and Dardouri created a large ensemble that integrates several deep learning and machine learning models including autoencoders, recurrent neural networks, Random Forests, and gradient boosting [9]. These papers continuously evidence how diversity of models can lead to the increase of the robustness of intrusion detection systems and a decrease in errors.However, most of the hybrid frameworks currently rely on voting or aggregation procedures to mix models either sequentially or at the decision level. The proposed HAEnRF system, however, adopts the feature-level fusion approach whereby a Random Forest classifier concatenates latent representations along with reconstruction errors of several autoencoders. [3],[4],[5],[12].

3. PROPOSED HAEnRF FRAMEWORK:

The HAEnRF framework is designed as a hybrid intrusion detection architecture that integrates unsupervisedrepresentation learning with supervised classification. The overall objective is to detect malicious networktraffic in cloud environments while maintaining flexibility and scalability.The framework consists of five main components:

1. Data preprocessing and feature preparation
2. Representation learning using multiple autoencoder models
3. Reconstruction error analysis
4. Feature fusion
5. Random Forest classification

Each component plays a specific role in transforming raw network traffic into meaningful representationsthat can be used for accurate intrusion detection.

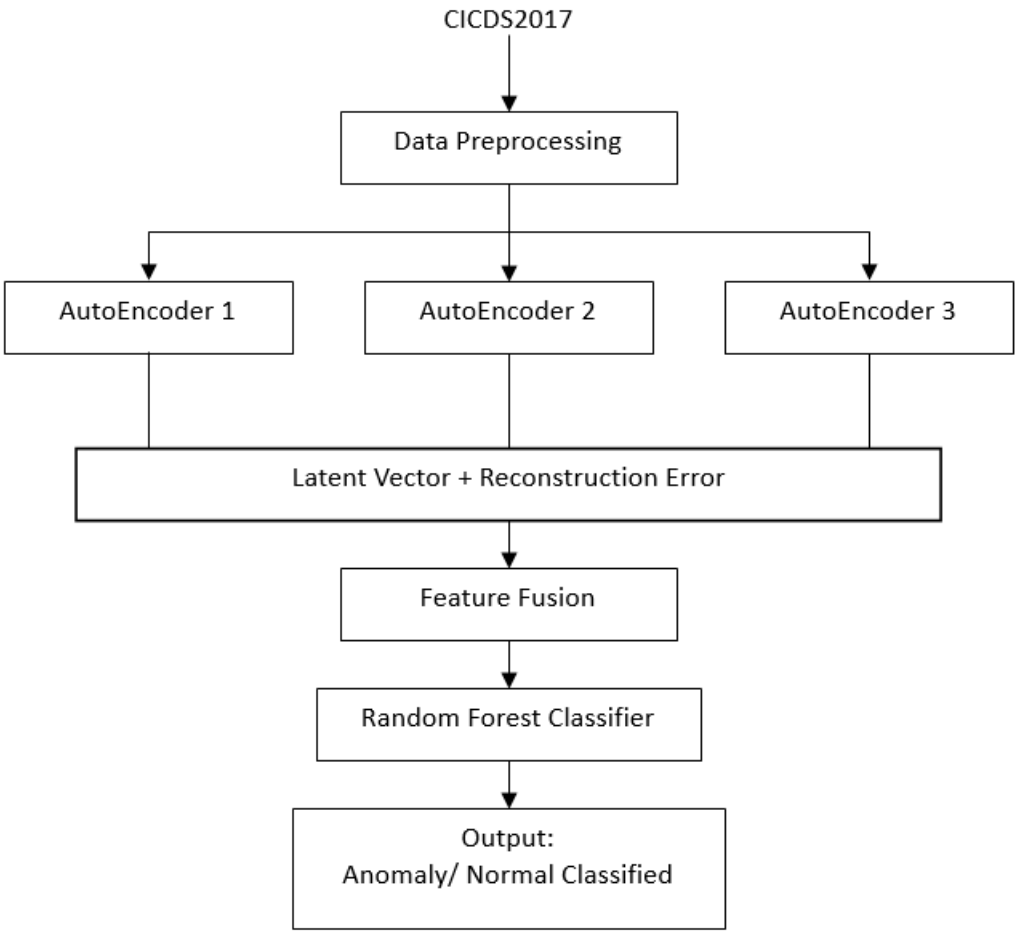

Figure 1 Proposed HAEnRF framework

3.1 Data Preprocessing

When we are dealing with network traffic datasets, we usually find that they have some missing information, repeated features, and categories that need to be cleaned up before we can use them to train our machine learning models. So, to make sure our data is good to go, we set up a step-by-step preprocessing pipeline that gets everything ready for model training. This way, we can trust that our data is accurate and consistent, which is crucial for getting reliable results from our models.First, irrelevant or constant features are removed to reduce dimensionality and eliminate unnecessary noise. Features containing missing or undefined values are either cleaned or replaced with appropriate numerical representations. Categorical attributes are transformed into numerical form using encoding techniques so that they can be processed by machine learning algorithms.To address potential class imbalance in the dataset, oversampling techniques are applied to increase the representation of minority classes. This step is particularly important in intrusion detection because attack samples often occur less frequently than normal traffic.To make sure everything works well, the numbers in the data are adjusted to a standard scale. This adjustment, called normalization, prevents features with big ranges of numbers from taking over when the model is being trained.

3.2 Autoencoder-Based Representation Learning

After preprocessing, the cleaned dataset is used to train three different autoencoder architectures. Each architecture is designed to capture different structural characteristics of network traffic data.

Dense Autoencoder

The dense autoencoder is made up of several fully connected layers that shrink the original feature space into a smaller, more compact form. This design works well for datasets that have tables with rows representing traffic flow, where each row has a set of numbers that describe it.

Denoising Autoencoder

The denoising autoencoder introduces controlled noise into the input data during training. The network learns to reconstruct the original clean input despite the presence of noise. This process encourages the model to learn robust features that generalize well to unseen traffic patterns.

Convolutional Autoencoder

The convolutional autoencoder applies one-dimensional convolution operations to capture relationships between neighboring features. Although convolutional architectures are commonly used for image processing, they can also reveal meaningful local interactions in structured network traffic data.By using multiple autoencoder architectures, the framework can learn diverse representations of the same network traffic patterns.

3.3 Reconstruction Error Analysis

When the autoencoders have finished learning, they try to rebuild the original information. To see how well they did, we compare the original and rebuilt data to find out the difference, and this is measured by how far off the rebuilt data is from the original, which is called the reconstruction error.Reconstruction errors provide useful signals for identifying anomalies. Traffic samples that differ significantly from normal patterns tend to produce higher reconstruction errors. Instead of relying solely on fixed thresholds to classify anomalies, these error values are incorporated into the feature fusion process.

3.4 Feature Fusion

The HAEnRF framework brings together different types of information from autoencoder models to create a more complete picture. This is done by combining several key elements, which are merged to form a single, unified feature set. Specifically, the framework integrates the following components which is

- latent representations produced by each encoder
- reconstruction error values from each autoencoder

These features are put together to make a single, longer feature vector for each sample of network traffic. This combined representation catches both the patterns in the structure that the encoders learned and the signals that show something is wrong, which come from errors in rebuilding the data.The resulting feature space provides richer information than the original dataset alone.

3.5 Random Forest Classification

The last part of the HAEnRF framework uses a special tool called Random Forest to find intrusions. This tool is a type of ensemble learning, which means it uses many decision trees to make a prediction. When it's trained, it builds many trees, and each tree makes its own guess. Then, it looks at all the guesses and chooses the most popular one as the final answer.Random Forest models offer several advantages for intrusion detection tasks. They are relatively resistant to overfitting, can handle high-dimensional feature spaces, and perform well even when features are correlated. Additionally, class weights can be adjusted to address imbalanced datasets.In this approach, the combined information from the autoencoder step is used as input for the Random Forest classifier. The classifier figures out how different mixes of hidden features and errors in reconstruction match up with normal or harmful traffic patterns. This helps the system learn and make better decisions about what's safe and what's not. The autoencoder's job is to find the most important features in the data, and the Random Forest classifier uses these features to make predictions. By working together, they can identify traffic patterns that might be malicious, and help keep the system secure.

1DConvolutional Autoencoder (Conv1DAE):

The third autoencoder uses a different inductive approach by applying one-dimensional convolution to the input feature vector. In this model, flow features are interpreted as a sequential structure rather than independent attributes. The encoder consists of stacked Conv1D layers with pooling operations (e.g., Conv1D with 32 filters → max pooling → Conv1D with 64 filters → max pooling) to capture local relationships between nearby features.These convolutional filters help detect short-range dependencies and feature interactions that dense layers may not easily capture. After convolution and pooling, the extracted feature maps are flattened and compressed into a 16-dimensional latent representation. The decoder reconstructs the input using convolutional layers with upsampling operations. ReLU activation functions are applied in hidden layers, and mean squared error (MSE) is used as the reconstruction loss.By learning structured patterns in flow-level data, the Conv1D autoencoder can identify correlated variations among related traffic features. Combined with dense and denoising autoencoders, this model provides an additional perspective for representing network traffic behavior[10].

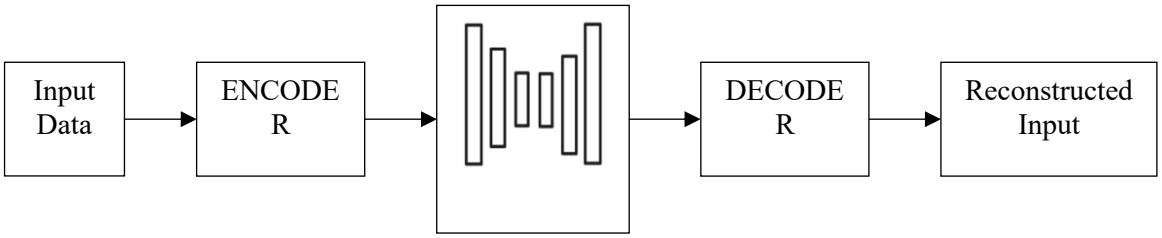

Figure 2: Diagram showing how an autoencoder works, starting from the input and ending with the creation of a 16D latent vector and input reconstruction

All three autoencoders are trained using the Adam optimizer to minimize the mean squared reconstruction error between the original input x and its reconstruction $\hat{x}$. Training is performed on the training subset, with reconstruction loss monitored on the validation set. Early stopping is used during training to halt the learning process when the validation performance stops improving, thereby preventing overfitting. Each autoencoder is defined by an encoder function $z = f_{\text{enc}}(x)$ which converts an input flow $x$into a 16-dimensional latent vector (z), and a decoder function $\hat{x} = f_{\text{dec}}(z)$that reconstructs the input. During inference, two outputs are generated for every sample$i$:

(a) the latent feature representation z_i

(b) the reconstruction error $e_i = \| x_i - \hat{x}_i \|^2$.

After training, all samples from the training, validation, and testing datasets are passed through each autoencoder to obtain these outputs. When we look at traffic flows, we can see that the ones that follow the usual patterns tend to have lower errors when we try to rebuild them. On the other hand, flows that are unusual or don't happen very often have higher errors. To deal with this, the HAEnRF framework uses three different autoencoder architectures that work together to capture the different aspects of network traffic. One of these autoencoders, the dense autoencoder, looks at the big picture and models the relationships between all the features. Another one, the denoising autoencoder, makes the system more robust by learning from inputs that have been corrupted on purpose. Then there's the Conv1D autoencoder, which focuses on the local correlations between features that are close to each other. By combining these, we get a 16-dimensional representation that's really useful for the later stages of fusion and classification. This helps us get a better understanding of the traffic flows and make more accurate predictions. The way these autoencoders work together is key to capturing the complexities of network traffic and improving our ability to analyze and understand it.

Table 1: Proposed Autoencoder model specifications

Component	Dense Autoencoder	Denoising Autoencoder	Conv1D Autoencoder
Input Dimension	d	D (with noise added during training)	d reshaped as a 1D feature sequence
Encoder Layers	[d, 256, 128, 64]	[d, 256, 128, 64]	Conv1D(32) → Pool → Conv1D(64) → Pool
Latent Dimension	16	16	16
Decoder Layers	[64, 128, 256, d]	[64, 128, 256, d]	Upsample→ Conv1D(32) → Upsample → Conv1D(1)
Output Dimension	d	d	d
Activation	ReLU / Linear	ReLU / Linear	ReLU / Linear
Loss Function	MSE	MSE	MSE
Training Strategy	Standard training	Noisy input training	Local feature correlation learning

The input flow features are compressed by each autoencoder into a fixed 16-dimensional latent representation.This design decision was aligned with very early experimental work in the field. Those studies showed that the dimension of the latent space was a major factor in the intrusion detection effectiveness. Expanding the latent space too much, meant the model lost the ability to compress efficiently and thus the

advantages of representation learning got minimized. On the other hand, very small latent spaces could lose the important information that is necessary for an accurate characterization of the network traffic patterns. [5]

Besides deriving the latent embeddings, every autoencoder also generates a reconstruction error. This metric reveals the level to which the model can replicate the original network flow. Hence the reconstruction errors double as the suspiciousness indication as well, since patterns of the traffic that are off the learned distribution usually result in bigger reconstruction discrepancies. So, by mixing these error numbers in with the latent representations, the system not only preserves the exclusively internalized patterns but also the anomaly-sensitive signatures.The last step in the representational learning is to put together final features.The latent vectors extracted from each autoencoder are then concatenated with their reconstruction errors to make a composite feature vector. This vector then serves as an input to a Random Forest classifier for supervised learning. Random Forest classifiers are well-suited for this task mainly because they work well with tabular datasets, they are quite robust to overfitting and they can effectively model complex interactions among a mix of features [3], [11]. Besides, during the training balanced class weighting is used to lessen the impact of class imbalance which is almost always a problem with network intrusion datasets.

Feature Fusion

After the three autoencoder models have been trained, feature fusion is the next part of the HAEnRF framework that executes. At this point, the results of each autoencoder are merged into a single representation that can be directly applied for the classification purposes. Instead of isolating the treatment of each model, the fusion attempts at integrating the different unsupervised views of the same network traffic record into a single feature space.Every autoencoder upon the given input flow sample will generate latent representation of 16 dimensions and one reconstruction error scalar. So, for each network traffic, one set of three latent vectors (one per each autoencoder) and three reconstruction error magnitudes are obtained. Then, these four elements are concatenated together to create one feature vector with the dimensionality of 3x16+3=51. During the supervised learning, the ground-truth class label is also added, resulting in a dataset where each instance is described by 51 input features. Rather than operating directly on the raw traffic attributes, the Random Forest classifier uses this fused representation.To this end, the fusion approach is inspired by the fact that the different architectures of the autoencoders capture separate aspects of the network traffic.

The reconstruction errors of each model act as a supplementary signal indicating the extent to which the observed traffic fits the normal patterns. In fact, it is likely that malicious traffic behaves inconsistently with different representations. Regarding the example, while the attack's global statistical properties may be in line with the normal ones, the localized feature dependencies might be changed. Under these circumstances, one autoencoder might be able to reconstruct the input very accurately whereas the other one will produce errors that are much higher.Maintaining latent embeddings as well as reconstruction errors both, the merged feature representation keeps details about the learned structure in addition to how confident we are about the reconstruction.

It has been demonstrated through experiments that combining descriptive latent features with deviation-based reconstructions such as reconstruction errors leads to better anomaly detection [5], [6]. HAEnRF is taking this idea a step forward by combining the outputs of multiple autoencoders, where each one is giving different complementary information about the same network flow.One of the great benefits of this approach is the removal of the requirement that anomaly thresholds need to be manually selected, which is a practice that is quite common in standalone autoencoder-based intrusion detection systems. Rather than interpreting the reconstruction errors strictly as decision boundaries that have been fixed, HAEnRF sees them as features that are trainable in the classification step. Thus, the Random Forest classifier is given the ability to identify the relative importance of each dimension of the latent space and each signal coming from a reconstruction as per the training data [7], [8].

In a sense, every one of these autoencoders is playing the role of a representation learning expert that only learns a few aspects of the traffic distribution. When their results are combined by concatenating them, these different perspectives become not only protected but also are made available to the downstream classifier to be used for finding the relationships among various representations. This is exactly what makes this method valuable for intrusion detection, because different attacks are likely to be revealed by different feature subsets or behavioral patterns [6], [12].The final fused representation includes 51 features which is a compromise between descriptive richness and computational efficiency. Even though this representation is a lot more informative than the original raw feature space, it is still quite small in size allowing ensemble classifiers like Random Forest to interact with it efficiently [3]. Also keeping all autoencoders the same latent dimensionality is an advantage that facilitates the integration and guarantees structural uniformity of the combined feature vector.The feature fusion step literally is the heart of the HAEnRF system. Through this stage, each network flow is converted into a multi-view point feature vector which incorporates knowledge learned through several subspace models. As shown in Figure 2, such fused representation is then used as input for the supervised

classification stage that follows.

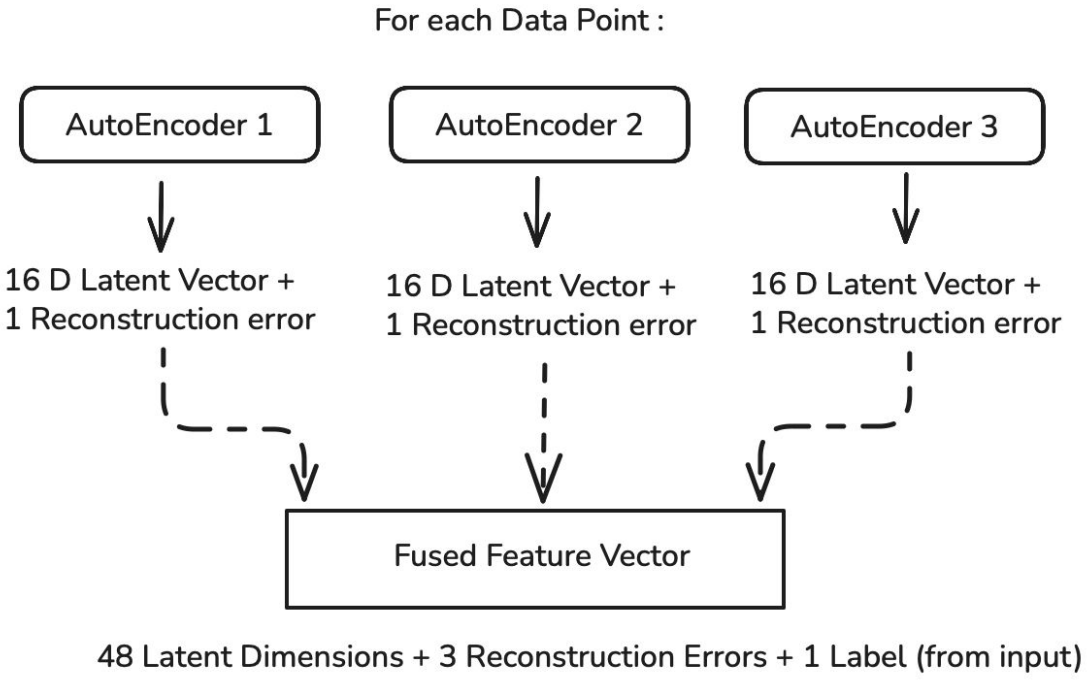

Figure 3 Formation of fused vector used for Random forest classifier training

Random Forest Classification

In the last phase of the HAEnRF framework, a Random Forest (RF) classifier is used to label network traffic flows based on a training dataset. Here, each flow can be classified into one of two groups: Benign or Attack. The classifier is fed by the hybrid feature representation that was created during the previous step. This single representation merges latent embeddings and reconstruction error scores from several autoencoder models. Hence, the classifier turns the anomaly cues discovered by unsupervised learning into a final decision about detection.The reason for choosing Random Forest is that it handles structured and tabular datasets very well - typical characteristics of network traffic data. Besides, RF models work efficiently with moderately high-dimensional feature spaces and they don't need a lot of feature engineering. On top of that, different features of network traffic are combining in complex, nonlinear ways. Random Forests are great at uncovering these patterns and, at the same time, they are resistant to noise and overfitting [3], [4]. All these features make the algorithm excellently adapted for learning from features produced by autoencoders, where the importance of features might vary depending on the type of attack.

A Random Forest model comprises several decision trees which are produced using the bootstrap aggregation (bagging) method. The idea is to create a tree on a randomly sampled subset of the training data during training. Moreover, when splitting a node in the tree, only a randomly selected subset of features is taken into account. This procedure helps to decorrelate trees and at the same time it increases the ability of the ensemble to generalize. Majority voting among all trees is used to determine the final prediction, which, in general, results in a decision boundary that is less prone to fluctuations than a single-tree one [3].The Random Forest in HAEnRF acts as a bridge between unsupervised representation learning and supervised classification. Each autoencoder is focused on different components of the network traffic patterns. As the Conv1D autoencoder may be highlighting burst-like temporal structures in traffic sequences, on the other hand, the errors of the denoising autoencoder may be going up when the attack behaviors are irregular or noisy. It is the Random Forest classifier which figures out how these different signals interact and collectively point to malicious activity. The key point is that this learning is done automatically during training and does not depend on manually set thresholds or handcrafted rules [7], [8].

The combination of the Random Forest parameters was established to find a good equilibrium between the ability of the model to represent data and its generalization potential. There are 200 decision trees in the ensemble which assist in diminishing the variance of prediction as well as snagging the stability up when dealing with features derived from neural networks. The maximal depth for any of these trees is set to 20 that limits the expressiveness of each tree by not allowing them to grow beyond a certain limit. Tree depth restrictions aid in the prevention of overfitting, that is memorizing the training samples, while still allowing the capturing of significant feature interactions.To deal with the problem of imbalanced classes, the balanced

class weighting method is used where the misclassification penalty for each class is scaled according to its occurrence in the training data. Apart from preprocessing and oversampling steps, there may still be some residual imbalance. The weighting method guarantees that minority attack samples have a sufficient impact on the learning process.To ensure reproducibility, a fixed random seed is set for model training. Besides, parallel computation is allowed so that several decision trees can be built at the same time, thereby boosting the computational efficiency. After this step, each network flow is labeled as either benign or malicious, with the latter category covering various types of attacks like botnet and DDoS. While the Random Forest demands labeled samples for its training, the autoencoders that were employed earlier in the processing done operate in a completely unsupervised fashion. This difference shows the hybrid character of the HAEnRF framework, enabling the system to detect anomalies even if those were not explicitly labeled during training.

4. EXPERIMENTALDESIGN AND PRELIMINARY EVALUATION

4.1 Experimental Setup

The experiment setup aims at testing the capability of the HAEnRF framework in a practical intrusion detection setting. The most important aim of this test is to find out how well the mixture of autoencoder-based representation learning and ensemble classification can perform on different types of traffic without being affected by class imbalance and the unpredictability of attack behaviors. All tests are performed with the CICIDS2017 dataset, which is considered a realistic reference point for intrusion detection studies [1]. This dataset includes usual network traffic and several attack types that were recorded over several days of network activity. Flow-level data feature a broad spectrum of behaviors, such as normal background traffic and DDoS attacks, botnet communication, and brute force login attempts. Due to such a variety, CICIDS2017 offers a perfect platform for evaluating intrusion detection systems designed for enterprise and cloud environments where network activity changes rapidly. The entire HAEnRF system is written in Python and leverages popular machine learning and deep learning libraries. The autoencoder networks are created with the help of mature deep learning tools like TensorFlow/Keras or PyTorch, while the Random Forest classifier makes use of the Scikit-learn library. These are relied upon for their dependability, reproducibility, and extensive use within the intrusion detection research community. All experiments run on a typical personal computer featuring a multi-core CPU, around 16-32 GB of RAM, and optionally a GPU for faster neural network training.After preprocessing, the dataset is split into three mutually exclusive subsets:

- Training set: 80%
- Validation set: 10%
- Testing set: 10%

Keeping the split the same helps to compare the results of the experiments and avoid accidental leakage of information. The autoencoder models are developed using only the training data in an unsupervised manner with Mean Squared Error (MSE) as the reconstruction loss function. Training takes place over many iterations and the stopping condition is a drop in the validation loss. Besides, to avoid overfitting, early stopping is applied once the validation performance no longer improves [5]. The validation data split is again employed in autoencoder model hyperparameter tuning such as the learning rate, batch size, network depth, and the level of noise in the denoising autoencoder. Meanwhile, the Random Forest classifier has been fitted using fused feature representations from the training data which is a quite common technique for ensemble methods. To ensure reproducibility, a fixed random seed is set for model training. Besides, parallel computation is allowed so that several decision trees can be built at the same time, thereby boosting the computational efficiency. After this step, each network flow is labeled as either benign or malicious, with the latter category covering various types of attacks like botnet and DDoS. While the Random Forest demands labeled samples for its training, the autoencoders that were employed earlier in the processing done operate in a completely unsupervised fashion. This difference shows the hybrid character of the HAEnRF framework, enabling the system to detect anomalies even if those were not explicitly labeled during training.Important RF parameters such as tree depth and classification thresholds are likewise fine-tuned via the validation set. This stringent division guarantees that the supervised and unsupervised parts get their respective optimizations without being exposed to the test data, therefore, the integrity of the evaluation process is maintained [3], [7].

Moreover, for achieving better stability and reduced variances of the obtained results, the concept of k-fold cross-validation on only the training dataset will be implemented in some of the experimental setups. Such a method not only mitigates over-dependence on a single train-test split but also provides the opportunity to further interrogate the models' stability during ablation studies and architectural differences examinations. Additionally, cross-validation sheds light on how sensitive a model is to changes in the training dataset and essentially leads to a more dependable estimation of performance consistency [4].One of the main evaluation points is the detection of unknown attacks by HAEnRF, which is a crucial feature of real-world intrusion detection systems. To recreate zero-day attack conditions, a few attack categories (for example, botnet traffic)

are left out deliberately from training data in an experimental set-up in a controlled manner. The models trained in such a way are tested on samples that contain these previously unseen attacks to check whether they are still able to pick up abnormal behavior by means of reconstruction error and learned feature interaction signals [6], [10].

Very careful measures to avoid data leakage were followed during the experiments. The test data was strictly off-limits for feature selection, model training and hyperparameter tuning. The computation of performance coefficients was performed only on the final model. In order to enhance the statistical validity and reduce the unevenness in the reported studies, the multiple experimental runs results obtained using different random data splitting techniques may be averaged [4], [12].

4.2 Evaluation Metrics

The set of conventional classification metrics that can be computed using a confusion matrix is utilized by the HAEnRF framework for evaluating the performance of the models. These metrics are also a standard in intrusion detection research as they reflect different properties of the model such as the ability to detect, reliability in classification and tolerance for class imbalance.Confusion matrix includes the following

- True Positives (TP): Malicious flows that were successfully recognized as attack
- True Negatives (TN): Benign flows that were correctly recognized as normal
- False Positives (FP): Benign flows that were wrongly classified as attacks
- False Negatives (FN): Attack flows that were wrongly classified as benign

The following evaluation metrics are all valid considering these four numbers.

Accuracy: Accuracy represents the proportion of correctly classified samples among all observations:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Although accuracy provides a general indication of model performance, it can be misleading in intrusion detection tasks where normal traffic significantly exceeds malicious traffic. For this reason, it should be interpreted together with other class-specific evaluation metrics [4].

Precision: Precision measures how reliable the predicted attack labels are by calculating the proportion of flows identified as malicious that are actually attacks:

$$Precision = \frac{TP}{TP + FP}$$

High precision is essential in practical security systems because frequent false alarms can burden analysts, increase operational effort, and reduce trust in automated detection tools [12].

Recall (Detection Rate): Recall evaluates the ability of the model to identify real attacks by measuring the proportion of actual malicious instances that are correctly detected:

$$Recall = \frac{TP}{TP + FN}$$

Recall is a critical metric for intrusion detection systems, since false negatives correspond to missed attacks that may lead to service disruption, data leakage, or prolonged compromise [4].

F1Score: It is calculated as the harmonic mean of precision and recall:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

For intrusion detection jobs, where both false positives and false negatives have serious operational ramifications, the F1 score is especially well suited. Extreme bias toward either metric is penalized by a single, balanced measure. [5]Besides relying on threshold-specific metrics, Receiver Operating Characteristic -Area Under the Curve (ROC-AUC) unfolding a deep insight into the comparison of model outputs on all possible decision thresholds when assessing classification performance is considered. ROC-AUC illustrates the trade-off between True Positive Rate (TPR) and False Positive Rate (FPR). Generally, Higher AUC scores signify that the two classes (benign and malicious) feature distributions are highly separable, and it depicts the degree of resistance to the effects of threshold choice. With this feature, the system can be turned into different operating modes depending on deployment circumstances and security requirements [11].HAEnRF will be assessed using all of these indicators in accordance with previous intrusion detection research in order to capture both attack detection efficacy and false alarm

behavior.[4,12]. Results are presented on a per-class basis (benign and attack) and utilizing macro averaged measures that give each class equal weight regardless of class frequency in order to encourage transparency and fairness. To learn more about the kinds of mistakes the model makes, the confusion matrix itself is analyzed in addition to scalar metric values. This analysis assists in determining whether certain traffic patterns are consistently misclassified and whether false positives or false negatives predominate in misclassifications. Additional metrics like balanced accuracy or the geometric mean (G mean) may be supplied to provide a more trustworthy evaluation across both classes in situations when residual class imbalance impacts interpretation. [4] .Following final model training and validation, all evaluation metrics are only calculated on the held-out test dataset. By doing this, it is ensured that published results accurately represent generalization performance rather than training data memorization.

4.3 Preliminary Results and Feasibility Analysis

To help establish the validity of the proposed HAEnRF system, thebaseline and comparative experiment results are presented in this section. The obtained results are a demonstration of the framework conceptual soundness and operational capability and not a claim of absolute state-of-the-art performance. The experiments are conducted using the evaluation protocol using CICIDS2017 dataset with a 80:10:10 train-validation-test split[1,11].

Comparative Models:

The following models are tested and compared:

- Raw Random Forest (Raw RF): a Random Forest classifier trained directly on the preprocessed original flow-level features, without any autoencoder-based representation learning
- Single Autoencoder + RF Variants: two-stage models wherein a Random Forest classifier is trained on the latent representation and reconstruction error obtained from a single autoencoder at a time (Dense AE, Denoising AE, or Conv1D AE)
- HAEnRF - (Proposed): The complete hybrid framework integrating fused latent representations and reconstruction errors from all three autoencoders.

All Random Forest models are built with 200 trees, a maximum depth of 20, and utilize balanced class weighting [3, 11]. The performance on the held-out test set is measured using standard intrusion detection metrics. Table 2 describes the performance classification of the evaluated models based on Accuracy, Precision, Recall, F1-Score and ROC-AUC. The following results in Table 2 are evaluated as the preliminary feasibility evaluation which is intended to demonstrate the operational viability of the proposed framework

Table 2 Baseline and Comparative Performance on CICIDS2017

Model	Accuracy (%)	Precision	Recall	F1-Score	ROC-AUC
Raw RF	97.8	0.976	0.971	0.973	0.984
Dense AE + RF	98.4	0.981	0.979	0.980	0.991
Denoising AE+ RF	98.5	0.982	0.981	0.982	0.993
Conv1D AE+ RF	98.6	0.984	0.983	0.984	0.994
HAEnRF (Proposed)	99.2	0.991	0.987	0.988	0.997

The results of the proposed model are based on a supervised classification which was a Random Forest trained on a fused feature space consisting of latent embeddings and reconstruction error signals extracted from dense, denoising, and Conv1D autoencoders, and evaluated on the held-out test split of the CICIDS2017 dataset. The findings show that when incorporating autoencoder-based representation learning, the intrusion detection performance is improved over a Random Forest trained on raw features only [5,6]. Each single-autoencoder hybrid variant surpasses the Raw RF baseline in every metric, which confirms the advantage of learned latent representations, and reconstruction error features.

Within the single-autoencoder models, the Conv1D AE + RF version exhibits somewhat better results, which is likely due to its capacity to identify local feature correlations in flow-level traffic data [5,10]. However, the complete HAEnRF framework reaches the best global results and consistently beats all baseline and single-autoencoder variants.The performance improvements that we have noted can be explained by feature-level fusion of diverse autoencoder outputs providing complementary information that leads to

classification stability and better separability of benign and malicious traffic [5,6,7,8]. Especially, the use of reconstruction error signals along with latent embeddings allows the Random Forest classifier to utilize structural representations as well as deviation-oriented cues, thus it does not need manually set anomaly thresholds.It should be pointed out that these are only initial results which mainly provide evidence of the feasibility of the proposed HAEnRF architecture [4, 12]. Further extensive benchmarking, attack-specific analysis, and cross-dataset evaluation have been planned as next steps to evaluate the model's robustness and generalization under different operational conditions.

5. FRAMEWORK ANALYSIS AND DESIGN DISCUSSION:

At this point in the research, the main goal is to explain and substantiate the HAEnRF framework rather than to showcase perfected experimental results. Nevertheless, there is still room to touch upon the expected performance aspects of the proposed system by pointing to the trends discovered in similar intrusion detection studies that unite ensemble classifiers with autoencoder-based representation learning.For example, many recent methodologies shed light on the capability of Random Forest classifiers in intrusion detection applications. Chen et al. described obtaining almost 99.9% accuracy for DDoS-related detection scenarios when Random Forest architectures were finely tuned and trained on standard datasets [19]. Similarly, Almuhanna and Dardouri showed that combining machine learning and deep learning in hybrid ensemble strategies can yield very high detection results for different types of attacks [9]. These results all hint that ensemble-based classifiers with the aid of well-representing features can offer top-notch performance when it comes to intrusion detection nowadays.

In light of these evidences, the HAEnRF model is intended to maintain excellent detection capabilities with consistent performance on the usual metrics. The system architecture integrates various autoencoders alongside a Random Forest Classifier allowing the extraction of different informative network traffic representations. Moreover, this architectural mixture within the representation learning results stage decreases the dependence on a single feature viewpoint only. As such, the overall design should provide great figures of precision and recall, probably going beyond 95% for normal and attack types of traffic. Against them single-model alternatives that might only be efficient in detecting the majority class of attacks, the combined representation feature vector in HAEnRF should be able to handle effectively both large-scale attacks like volumetric DDoS traffic and the more stealthy or infrequent ones which leave weaker statistical signatures. In such scenarios one can reasonably expect F1-score values around the range 0.97–0.99 that indicate a good level of balance between false alarms and missed detections [5], [7].

Besides focussing simply on maximizing one particular evaluation criterion, HAEnRF was devised so as to yield balanced classification outcomes across both traffic classes. Bias in results can be an issue in real intrusion detection setups. On one hand systems producing lots of false positives can exhaust the analysts and hamper normal business operations; on the other hand high false negatives levels mean that the attacker manages to go through unflagged. Therefore, what one wants is to have a confusion matrix where True Positive and True Negative are on average high ensuring that the identified attack performance of the system con-incides with the one on legitimate network activities. Getting this sort of balance becomes more critical under cloud where diversity in traffic patterns occurs and one has limited tolerance for errors [4], [12].Another proof that a detector works well might be visible from the Receiver Operating Characteristic (ROC) aspect. If the ROC-AUC number is near 1 or even above that level, it will signal how well one can differentiate benign versus malicious throughout the whole detection threshold range.This feature is particularly useful when deploying systems in practice since operational security policies usually require the change of thresholds depending on acceptable false alarm levels and the willingness to take risks. A system that performs well over different thresholds is more versatile for being used in various security scenarios [11]. Besides, it is quite possible that the robustness of the proposed framework also comes from the integration of autoencoders in the pipeline which is mostly supervised. Hence, the Random Forest classifier learns on labeled data, at the same time, the reconstruction errors that come from the autoencoders are not directly linked to certain attack labels.

Rather, they are indicative of changes in the patterns that the model has learned during training sessions. Therefore, if new or changing attack vectors lead to traffic patterns that are different from the baseline, there is a chance that the reconstruction errors will increase even before the classifier is retrained. This means that HAEnRF might be able to spot a sign of a new threat and if allowed, the model can get used to new threats over time with the help of new labeled data. This capability is especially crucial when the system is left without human intervention for a long time in a changing cloud environment where network traffic is always changing [5], [6].To sum up, while the physical proof of the idea will be carried out in later stages of the implementation, the design of HAEnRF and what has been documented in the related studies lead to a belief of accurate detection, well-balanced performance classification, and huge generalization capability. It is these features that make HAEnRF a very promising hybrid method for intrusion detection in a cloud-oriented environment.

5.1 Planned Comparative Analysis

In order to get a deeper understanding on the effectiveness and unique characteristics of the HAEnRF system, a formal and organized comparative evaluation with a number of representative intrusion detection baselines is on the agenda for future work. What is expected from this paper is an analysis of the differences in numerical performance, as well as learning how the feature-level fusion strategy proposed impacts detection behavior in relation to more traditional or simplified approaches.

Raw Random Forest Baseline: The first model that is used as a baseline in the comparative analysis is a Raw Random Forest (Raw RF) classifier that is trained on the original flow-level network features after the standard preprocessing.This baseline excludes any form of autoencoderbased representation learning. Random Forests are widely used in intrusion detection due to their robustness and interpretability [3], [11]. Comparison with this baseline allows isolation of the effect of learned latent representations and reconstruction error features. Differences observed in future evaluations would help clarify whether the proposed representation learning stage contributes additional discriminatory information beyond raw statistical features.

SingleAutoencoder IDS Baselines: The second comparison group consists of intrusion detection models based on a single autoencoder architecture. This includes traditional anomaly detection approaches that rely on reconstruction error thresholding, as well as hybrid variants in which a Random Forest classifier is trained using the latent representation and reconstruction error from one autoencoder at a time. Dense, denoising, and Conv1D autoencoder are considered independently under this setting. These baselines are important for assessing whether any observed advantages arise from the inclusion of the autoencoder derived features in general ot from the the deliberate fusion of multiple heterogeneous autoencoder representations [5], [10].

Hybrid IDS Comparisons: The third set of comparison models consists of earlier proposed hybrid intrusion detection systems that combine autoencoders with Random Forest classifiers using different architectural mechanisms. In some studies, these components are implemented in sequential or multi-stage pipelines in which a Random Forest carries out the initial classification step and then an autoencoder refines the detection result. Other methods use Random Forest algorithms primarily for feature selection and after that the reduced feature set is fed to the autoencoder-based anomaly detection module [7], [8]. These hybrid systems will be reimplemented during experimental phases as far as possible for obtaining fair evaluation results. In circumstances where reimplementation is not doable, the performance results from literature will be used as contextual points of reference instead of being treated as direct quantitative benchmarks. Figure 4 depicts a conceptual illustration of the relative behavior patterns expected among these different model categories. The purpose of this figure is purely to illustrate and it tries to figure out the expected relationships inferred from the intrusion detection research rather than presenting the experimental findings in the paper.

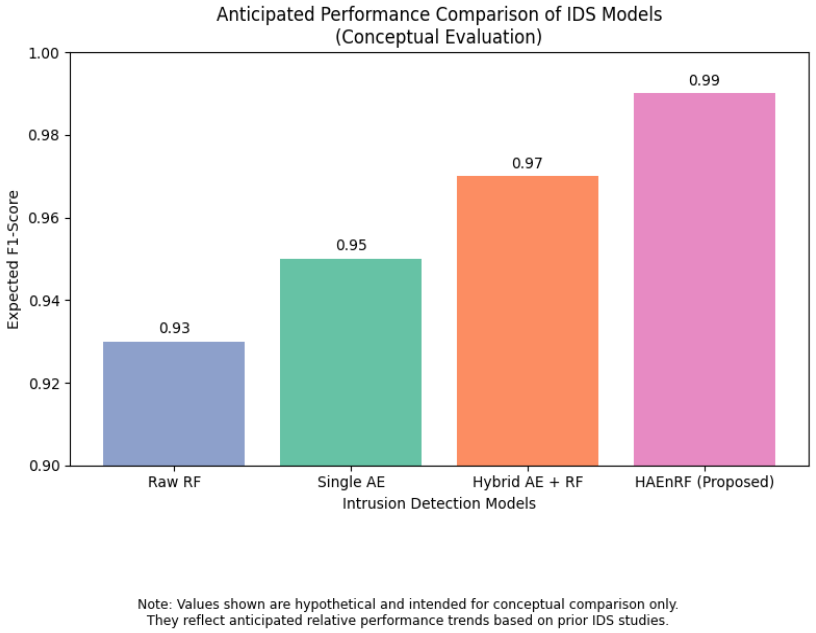

Figure 4 Conceptual comparison of expected performance trends among IDS model categories based on prior studies

Furthermore, feature importance measures derived from the Random Forest component of HAEnRF can be examined to understand the relative contributions of latent dimensions and reconstruction error features from different autoencoders. Such analysis supports interpretability and helps validate the rationale for feature-

level fusion within the proposed framework.

6. CONCLUSION AND FUTURE WORKS

In this paper, the proposed HAEnRF is a hybrid intrusion detection system, which combines multiple autoencoder based unsupervised representation learners with a supervised Random Forest classifier for cloud environment flow-level attack detection. Through integrating dense, denoising, and one-dimensional convolutional autoencoders, the mechanism is able to acquire diverse features of network traffic independently and combine latent representations with reconstruction error signals at the feature level. Such a layout allows the incorporation of anomaly-sensitive information effectively without the need for manually defined thresholds.The presented structure is attack and cloud agnostic since it works on flow-level features and can be re-trained on different datasets without changing its architecture. The paper's content mainly revolves around developing and justifying the HAEnRF approach, with an experimental evaluation at the very first level of feasibility using the CICIDS2017 dataset to support that. The outcome of the study suggests shows the feasibility of the approach, and the authors do not intend to use their results as a basis for claiming performance.The study plans to carry out experimental benchmarking on different datasets, perform attack-specific analysis, and assess the system in real-time deployment scenarios such as with low latency and limited computational capacity. Some other areas of investigation could be temporal modeling, dealing with concept drift by using incremental learning techniques, and enhancing model interpretability to help facilitate their use in operational security environments.

References

1. Sharafaldin, Iman, Arash Habibi Lashkari, and Ali A. Ghorbani. "Toward generating a new intrusion detection dataset and intrusion traffic characterization." *ICISSp* 1, no. 2018 (2018): 108-116.
2. Chawla, Nitesh V., Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. "SMOTE: synthetic minority over-sampling technique." *Journal of artificial intelligence research* 16 (2002): 321-357.
3. Breiman, Leo. "Random forests." *Machine learning* 45, no. 1 (2001): 5-32.
4. Buczak, Anna L., and Erhan Guven. "A survey of data mining and machine learning methods for cyber security intrusion detection." *IEEE Communications surveys & tutorials* 18, no. 2 (2015): 1153-1176.
5. Song, Youngrok, Sangwon Hyun, and Yun-Gyung Cheong. "Analysis of autoencoders for network intrusion detection." *Sensors* 21, no. 13 (2021): 4294.
6. Mirsky, Yisroel, Tomer Doitshman, Yuval Elovici, and Asaf Shabtai. "Kitsune: an ensemble of autoencoders for online network intrusion detection." *arXiv preprint arXiv:1802.09089* (2018).
7. Wang, Chao, Yunxiao Sun, Wenting Wang, Hongri Liu, and Bailing Wang. "Hybrid intrusion detection system based on combination of random forest and autoencoder." *Symmetry* 15, no. 3 (2023): 568.
8. Li, XuKui, Wei Chen, Qianru Zhang, and Lifa Wu. "Building auto-encoder intrusion detection system based on random forest feature selection." *Computers & Security* 95 (2020): 101851.
9. Almuhanna, Reem, and Samia Dardouri. "A deep learning/machine learning approach for anomaly based network intrusion detection." *Frontiers in Artificial Intelligence* 8 (2025): 1625891.
10. Park, Hyoseong, Dongil Shin, Chulgyun Park, Jisoo Jang, and Dongkyoo Shin. "Unsupervised machine learning methods for anomaly detection in network packets." *Electronics* 14, no. 14 (2025): 2779.
11. Chavan, Prathamesh V., and Nilesh V. Alone. “Optimizing Intrusion Detection with Random Forest: A High-Accuracy Approach Using CIC-IDS 2017.” *International Journal of Computer Applications* 187, no. 3 (2025):0975-8887.
12. Pinto, Andrea, Luis-Carlos Herrera, Yezid Donoso, and Jairo A. Gutierrez. "Survey on intrusion detection systems based on machine learning techniques for the protection of critical infrastructure." *Sensors* 23, no. 5 (2023): 2415.
13. Moustafa, Nour, and Jill Slay. "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)." In *2015 military communications and information systems conference (MilCIS)*, pp. 1-6. IEEE, 2015.
14. Shone, Nathan, Tran Nguyen Ngoc, Vu Dinh Phai, and Qi Shi. "A deep learning approach to network intrusion detection." *IEEE transactions on emerging topics in computational intelligence* 2, no. 1 (2018): 41-50.
15. Yin, Chuanlong, Yuefei Zhu, Jinlong Fei, and Xinzheng He. "A deep learning approach for intrusion detection using recurrent neural networks." *Ieee Access* 5 (2017): 21954-21961.
16. S. Kim, S. Kim, and J. Yoo, “A Lightweight and Robust Deep Learning System for Intrusion Detection,” *Proceedings of the IEEE Network Operations and Management Symposium (NOMS)*, 2016.
17. K. Hodo, X. Bellekens, A. Hamilton, R. Broadhurst, and Y. Sekercioglu, “Intelligent Intrusion Detection System Using Deep Neural Networks,” *IEEE Transactions on Network Science and Engineering*, vol. 5, no. 2, 2018, pp. 1153–1162.
18. Hussain, Fatima, Rasheed Hussain, Syed Ali Hassan, and Ekram Hossain. "Machine learning in IoT security: Current solutions and future challenges." *IEEE Communications Surveys & Tutorials* 22, no. 3 (2020): 1686-1721.
19. Chen, Shao-Rui, Shiang-Jiun Chen, and Wen-Bin Hsieh. "Enhancing Machine Learning-Based DDoS Detection Through Hyperparameter Optimization." *Electronics* 14, no. 16 (2025): 3319.