Advancing Network Anomaly Detection: An Ensemble Approach Combining Optimized Contractive Autoencoders and K-Means Clustering

Sharmin Aktar
Department of Computer Science
University of New Orleans
New Orleans, LA, USA 70148
Email: saktar@uno.edu

Abdullah Yasin Nur
Department of Computer Science
University of New Orleans
New Orleans, LA, USA 70148
Email: ayn@cs.uno.edu

***Abstract*—This paper introduces a new approach for detecting unusual activities in network traffic, a critical aspect in maintaining network security. We propose an innovative model that combines the strengths of Contractive Autoencoders (CAEs) and K-means clustering, specifically designed for effective anomaly detection in network environments. Our model employs CAEs for efficient data processing and K-means clustering to identify deviations from standard network patterns. The focus is on the exploration of CAE's latent space and the impact of various deep learning parameters on the model's detection capabilities. Tested on the NSL-KDD dataset, a standard in network security research, our best-tuned model achieves an F1 Score of 0.92, making it approximately 8.2% more effective than the basic Autoencoder model and about 5.7% better than the standalone K-Means approach in terms of F1 Score. This significant improvement in performance highlights the advanced capabilities of our model in identifying potential threats in network traffic, marking a considerable advancement in the field of network security.**

***Index Terms*—Network Security, Anomaly Detection, Deep Learning, Autoencoder, Contractive Autoencoder, K-means Clustering, Network Traffic Analysis, Intrusion Detection, CAE-K-means Model, Cybersecurity**

I. Introduction

Network security is a vital aspect of the modern digital world but is threatened by the continuous development of complex cyberattacks. Major modern cyber threats include denial of service (DoS) and its distributed form (DDoS), which focus on disrupting network availability by flooding with traffic [3, 8]. Traditional techniques for identifying abnormalities in network traffic, while foundational, are increasingly insufficient against these evolving threats. This inadequacy highlights the need for more advanced detection methods. In this work we address this need by presenting a novel approach that integrates Contractive Autoencoders (CAEs) with K-means clustering.

Our goal is to develop a more robust network security system by creating a novel technique for detecting anomalies in network data that is both more accurate and efficient. Our method first leverages CAEs to identify standard traffic patterns, simplifying the network data. We then apply K-means clustering on this processed data to pinpoint deviations that may indicate anomalies. A key aspect of our study is examining the CAE's latent space - the compressed data representation - and how its configuration can enhance anomaly detection. Furthermore, we explore how different deep learning parameters impact the performance of detecting network intrusions. In summary, our approach combines CAEs and K-means clustering in an innovative way to improve network security against sophisticated modern threats.

The main contributions of this paper are summarized as follows:

- Developing a new anomaly detection model that combines Contractive Autoencoders with K-means clustering, specifically designed for network traffic analysis.
- Exploring the CAE's latent space and deep learning hyperparameters to improve anomaly detection performance.
- Evaluating the model's performance on the NSL-KDD dataset, demonstrating superior accuracy over existing techniques.

The rest of the paper is structured as follows. Section II reviews various network attack detection approaches. Section III provides our methodology. Section IV presents our results. Lastly, Section V concludes the paper.

II. Related Work

Recent developments in anomaly detection have seen major contributions, especially in exploring autoencoder latent spaces. This section highlights key studies aligned with our research's focus on utilizing autoencoders' latent space, particularly Contractive Autoencoders (CAE), to identify anomalies.

Erfani et al. [7] investigate a hybrid model combining deep autoencoders with One-Class SVMs for detecting anomalies in network traffic. This demonstrates a novel approach for discerning between normal and anomalous traffic patterns.

Sakurada and Yairi [12] explore using autoencoders for anomaly detection in complex systems. They show how autoencoders can efficiently handle high-dimensional data and identify anomalies based on reconstruction errors, which is highly relevant to our latent space analysis.

Moreover, Zhai et al. [10] experiment with deep structured energy-based models for anomaly detection, offering insights into adapting deep learning, especially autoencoders, to pinpoint irregular patterns in datasets.

In our previous research [11], we made contributions to this rapidly progressing field by developing a deep learning model that utilized a contractive autoencoder for DDoS attack detection. Our focus was on training the model to differentiate between normal and anomalous network traffic based on their compressed representations in the latent space, subsequently applying a stochastic threshold method to detect attacks. We evaluated our model on well-established datasets showing improved performance compared with the other existing models.

These prior studies establish the foundation for our current research by demonstrating various techniques for utilizing autoencoders for anomaly detection. Our present work aims to advance these approaches further by focusing on optimizing the latent space of contractive autoencoders, enabling more effective anomaly detection.

III. Methodology

Our anomaly detection approach in network traffic integrates two key machine learning concepts: Autoencoders and K-means clustering. This section outlines the theoretical framework and practical implementation of these concepts in our study.

A. Autoencoder

An autoencoder is an unsupervised neural network used for learning efficient data codings. Its architecture has two primary components: an encoder and a decoder. The encoder compresses the input into a lower-dimensional latent space, while the decoder reconstructs the original input from this compressed representation. By attempting to replicate the input, the autoencoder learns to capture its most salient features [13].

The autoencoder aims to minimize reconstruction error, typically measured by mean squared error between the input and output. Mathematically, its objective function is:

$$J_{AE}(\theta) = \frac{1}{n}\sum_{i=1}^{n} ||x_i - x_i'||^2 \tag{1}$$

where n is the number of data points, x_i is the ith input, and x_i' is the ith reconstructed input. By optimizing this objective, the autoencoder learns an efficient compressed representation of the input for effective coding and reconstruction.

B. Contractive Autoencoder

A Contractive Autoencoder (CAE) modifies the standard autoencoder by adding a regularization term to the loss function. This regularization term is the Frobenius norm of the Jacobian matrix of the encoder activations with respect to the input. Adding this term enhances the robustness of the learned representations by making them invariant to small variations in the input data [14].

The objective function for a CAE is:

$$J_{CAE}(\theta) = \frac{1}{n}\sum_{i=1}^{n} ||x_i - x_i'||^2 + \lambda||J_f(x)||_F^2, \tag{2}$$

where λ is a hyperparameter that controls the tradeoff between reconstruction accuracy and robustness of the extracted features. By including this additional regularization penalty, the CAE learns latent representations that are more robust to minor input perturbations.

C. K-means Clustering

K-means is a widely used clustering algorithm in machine learning that partitions data points into K distinct clusters based on feature similarity. The algorithm is designed to minimize the within-cluster variance by assigning each point to the nearest of the K clusters [5]. This makes it well-suited for detecting outliers or anomalies which are distant from cluster centers.

D. Integrated Method for Anomaly Detection

Our integrated anomaly detection approach uses Contractive Autoencoders (CAEs) and K-means clustering to improve performance on network traffic data. The method first trains a CAE on normal network data, enabling it to learn standard traffic patterns in the latent space. K-means clustering is then applied in this latent space to differentiate between regular and anomalous data points.

Additionally, we tune CAE hyperparameters including batch size, hidden layers, regularization, and learning rate to optimize anomaly detection capability. We also implement a technique to find the best threshold for identifying anomalies using K-means distance metrics. Through evaluation, the threshold maximizing F1 score is selected for accurate anomaly detection.

Extensive benchmarking confirms the reliability and accuracy of this integrated approach for network anomaly detection, outperforming conventional methods. In summary, our technique combines the strengths of deep learning and clustering to improve anomaly detection.

The algorithm 1 briefly summarizes the key steps of our anomaly detection method. It first trains a CAE on normal traffic to learn standard patterns (line 6). The CAE encoder extracts latent representations for train, validation, and test data (lines 7-9). Optimal threshold is determined by maximizing validation F1 score (lines 10-12). This threshold classifies test data by comparing latent representation distances to clusters (lines 13-14), and performance is evaluated against test labels (lines 15).

IV. Experiments

In this section, we first summarize the datasets utilized for our experiments. After that, we explain the evaluation metrics of our proposed model's performance, detailing our experimental setup. Finally, we show the experimental results of our proposed model and compare them with the related works.

Algorithm 1 Integrated Anomaly Detection with CAE and K-means

procedure INTEGRATED_ANOMALY_DETECTION(X_{train}, X_{val}, X_{test}, y_{val}, y_{test})
  Input:
    X_{train}, X_{val}, X_{test} = Network traffic data
    y_{val}, y_{test} = Data labels
  CAE_{model} = TUNE_CAE_PARAMS( ) ▷ Tune hyperparameters
  $trained\_CAE$ = TRAIN_CAE(CAE_{model}, X_{train}) ▷ Train on normal data
  Z_{train} = Encode(X_{train}) ▷ Get latent features
  Z_{val} = Encode(X_{val})
  Z_{test} = Encode(X_{test})
  $clusters$ = KMeans_Clustering(Z_{train})
  $distances_{\text{val}}$ = DISTANCE($Z_{\text{val}}, clusters$)
  $optimal\_threshold$ = FIND_OPTIMAL_THRESHOLD($distances_{\text{val}}, y_{\text{val}}$)
  $distances_{\text{test}}$ = DISTANCE($Z_{\text{test}}, clusters$)
  y_{pred} = CLASSIFY($distances_{\text{test}}, optimal\_threshold$)
  $metrics$ = EVALUATE($y_{\text{test}}, y_{\text{pred}}$)
  return $metrics$

A. Dataset Description and Preprocessing

For evaluating our model, we leveraged the NSL-KDD dataset [2], an improved version of the widely used KDD Cup '99 dataset. The NSL-KDD dataset addresses the class imbalance and redundancy limitations present in the original KDD data [4]. It contains a range of network intrusions along with normal traffic, making it suitable for benchmarking real-world network intrusion detectors. Specifically, the NSL-KDD dataset includes various attack types - denial-of-service, probing, remote-to-local, user-to-root, and benign traffic collected over TCP/IP. It has 41 features related to connection duration, protocols, and traffic and various traffic characteristics [6]. We performed the following preprocessing steps for effective training of our model:

- **Feature Selection:** Removed non-essential features, such as socket information (Source/Destination IP, Ports, and Flow ID) to avoid overfitting and improve model generalizability [11].
- **One-hot Encoding:** Encoded categorical features into numerical form through one-hot encoding. This approach expanded the feature set from 41 to 121 numeric features, enhancing the model's ability to distinguish between different traffic types.
- **Class Label Encoding:** Encoded the class labels into binary values, with '1' representing anomalies and '0' representing normal traffic, aligning with the binary classification approach of our model.
- **Duplicate Removal:** Removed duplicate records to avoid training bias, ensuring equal representation of each pattern [11].
- **Data Cleaning:** Filtered samples with NaN or INF values to maintain data integrity and quality.
- **Normalization:** Normalized numeric features to [0,1] range using min-max normalization.
- **Constant Feature Removal:** Removed features with constant values across all samples to reduce dimensionality without losing significant information.

B. Experimental Setup

We employed a comprehensive evaluation approach for our anomaly detection model using the NSL-KDD dataset, encompassing the computational environment, model configuration, training strategy, and assessment techniques.

1) Computational Environment: The experiments were conducted in PyTorch, enabling efficient neural network operations. The hardware comprised an Intel Core i7 processor and 16GB RAM for rapid data processing.

2) Model Configuration: The model architecture was a contractive autoencoder, chosen for its robust feature learning capabilities to capture normal network behavior patterns. The input layer corresponded to the 121 preprocessed NSL-KDD features. One hidden layer, referred to as the latent layer, was used to learn an optimized compressed representation. The output layer reconstructed inputs for compact representation learning.

3) Training Strategy: We divided the NSL-KDD dataset into three subsets: 70% for training, 15% for validation, and 15% for testing. The training set included only normal traffic to align with the one-class classification approach. The model was trained to minimize the reconstruction error, essential for learning the normal behavior in network traffic.

4) Hyperparameter Tuning: To optimize our model's performance, we conducted a two-phase hyperparameter tuning process. Initially, we explored a limited range of key hyperparameters including batch size, hidden dimensions, lambda regularization values, and learning rate. Different combinations of these parameters were iteratively evaluated to fine-tune the Contractive Autoencoder. Specifically, batch sizes of {32, 64, 128, 256}, hidden dimensions of {32, 64}, initial lambda values of {0.0001, 0.001, 0.01}, and learning rates of {0.001, 0.0001} were tested.

After identifying the most promising lambda value from this initial set, we further expanded our analysis to a broader range of lambda values. This extended exploration of regularization strength was critical to fully understand its impact on our model's capabilities and determining the optimal configuration.

5) Evaluation and Validation: We utilized multiple performance metrics beyond just accuracy. This provides a more comprehensive assessment, which is important for imbalanced anomaly datasets where anomalies occur much less frequently than normal samples [1]. With such skewed distributions, accuracy alone can be misleading if models achieve high accuracy by primarily classifying the majority normal data correctly. Our multi-metric approach enables properly evaluating the model's ability to identify rare anomalies, not just accurately

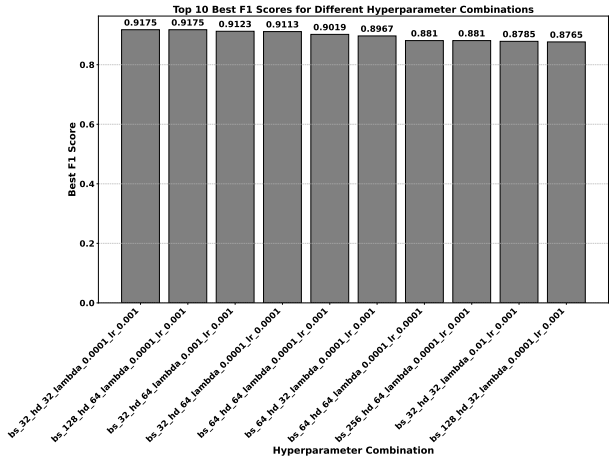

Fig. 1: Comparison of Model Performance Across Different Hyperparameter Combinations

TABLE I: Hyperparameters Used for Model Tuning

Hyperparameter	**Range of Values**
Batch Sizes	32, 64, 128, 256
Hidden Dimensions	64, 32
Learning Rates	0.001, 0.0001
Lambda (Contractive Loss Coefficient)	0.0001, 0.001, 0.01

classify predominant normal data. This ensures rigorous validation of real-world anomaly detection capabilities.

- **Performance Metrics**: Our key indicators included accuracy, precision, recall, F1-score, and PR-AUC.
  - *Precision*: Defined as the proportion of correctly identified positive instances out of all predicted positive instances. Calculated as Precision $= \frac{\text{TP}}{\text{TP+FP}}$. High precision indicates a low rate of false alarms.
  - *Recall*: Also known as sensitivity, it measures the proportion of actual positives that are correctly identified. Calculated as Recall $= \frac{\text{TP}}{\text{TP+FN}}$.
  - *Accuracy*: Measures the proportion of total predictions that are correct. Calculated as Accuracy $= \frac{\text{TP+TN}}{\text{TP+FP+TN+FN}}$.
  - *F1-score*: Harmonic mean of precision and recall, providing a balance between them. Calculated as F1-score $= 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision+Recall}}$.
  - *PR-AUC*: The area under the precision-recall curve. It is important in imbalanced datasets as it focuses on the performance of the model on the minority class, offering insight into how well the model detects actual positives.
- **Validation**: We regularly validated the model on a dedicated set to fine-tune its performance and reduce overfitting.
- **Testing**: The final assessment involved both normal and anomalous samples in the test set, examining the model's generalization and anomaly detection capabilities.

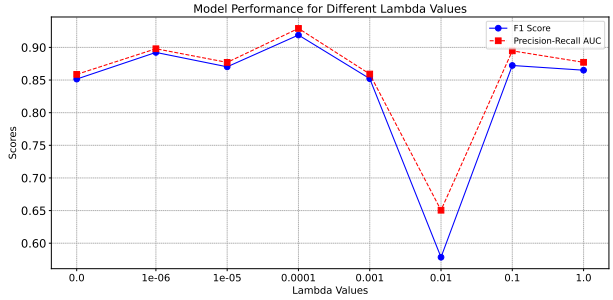

Fig. 2: Impact of Varying Lambda Values on Model Performance

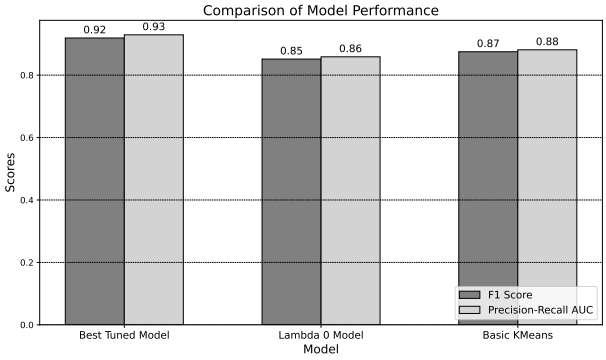

Fig. 3: Comparison of Model Performance

C. Results and Discussion

This subsection presents and discusses the results of our experiments on the NSL-KDD dataset using the proposed anomaly detection model. The findings are analyzed in the context of hyperparameter tuning, the impact of varying lambda values, and the comparative performance of our model against other methods.

1) Hyperparameter Tuning Analysis: As indicated in Table I and Figure 1, extensive hyperparameter tuning was conducted. The optimal combination was identified as a batch size of 32, hidden dimension of 32, lambda value of 0.0001, and a learning rate of 0.001. This combination resulted in the most effective learning and generalization capabilities for the model. Smaller batch sizes were found to be beneficial in terms of model accuracy and training efficiency, while the selected hidden dimension allowed for efficient feature extraction without overfitting.

2) Impact of Lambda on Model Performance: Our experiments evaluated various lambda settings in the contractive autoencoder to determine the optimal value for anomaly detection, as shown in Figure 2. Lambda controls the regularization strength, which is critical for model tuning. This figure illustrates the impact of different lambdas on the F1 score and Precision-Recall AUC, important metrics for assessing anomaly detection. Through experimentation, a lambda value of 0.0001 achieved peak results, with the model reaching an F1 score of 0.9189 and Precision-Recall AUC of 0.9291,

as depicted by the scores in Figure 2. This demonstrates the model's enhanced accuracy and reliability in identifying network anomalies at the optimal lambda setting.

3) Comparative Model Performance: In the comparative analysis, our model's efficacy, employing an optimized lambda parameter in a contractive autoencoder, was benchmarked against two baseline models. The first baseline is a standard autoencoder with no contraction penalty, denoted as the lambda 0 model, and the second is the conventional K-means clustering algorithm. The objective was to discern whether the latent representations derived from the contractive autoencoder could enhance the K-means algorithm's capability to segregate anomalous patterns from network traffic.

Referencing Figure 3, the integration of K-means with the contractive autoencoder's latent space yielded superior anomaly detection performance compared to the standalone models. This approach leverages the autoencoder's proficiency in capturing critical features, thereby facilitating a more precise clustering by the K-means algorithm.

The findings validate the hypothesis that the enhanced feature sensitivity introduced by the contractive autoencoder significantly bolsters the K-means algorithm's ability in differentiating between normal and abnormal network behaviors, a level of discrimination not achieved by using a basic autoencoder or K-means alone.

V. Conclusions

In this paper, we have presented a new model integrating Contractive Autoencoders and K-means clustering for detecting anomalies in network traffic. Our approach was thoroughly evaluated on the NSL-KDD dataset. Experiments optimizing the CAE's lambda hyperparameter identified 0.0001 as the optimal value, achieving peak F1 and AUC scores of 0.9189 and 0.9291 respectively. This demonstrates the model's enhanced accuracy and reliability for network anomaly detection when tuned properly. Comparative analysis showed our integrated method outperforms standard autoencoders and K-means alone. The results highlight the potential of combining deep learning and clustering techniques for more effective network security solutions.

References

[1] T. Hasanin, T.M. Khoshgoftaar, J.L. Leevy, and R. A. Bauder, "Severely Imbalanced Big Data Challenges: Investigating Data Sampling Approaches", Journal of Big Data, vol. 6, no. 107, 2019

[2] NSL-KDD Dataset, https://www.unb.ca/cic/datasets/nsl.html

[3] S. Aktar and A. Y. Nur, "Hash Based AS Traceback against DoS Attack", IEEE International Conference on Advanced Communication Technologies and Networking (CommNet), 2021

[4] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set", IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2009

[5] J. B. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations", Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Vol. 1, No. 14, pp. 281–297, 1967

[6] H. Choi, M. Kim, G. Lee, and W. Kim, "Unsupervised Learning Approach for Network Intrusion Detection System Using Autoencoders", The Journal of Supercomputing, Vol. 75, pp. 5597-5621, 2019

[7] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-Dimensional and Large-Scale Anomaly Detection using a Linear One-Class SVM with Deep Learning", Pattern Recognition, Vol. 58, pp 121-134, 2016

[8] A. Y. Nur and M. E. Tozal, "Record Route IP Traceback: Combating DoS Attacks and the Variants", Elsevier Computers & Security, Vol. 72, pp. 13-25, 2018

[9] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A Deep Learning Approach for Network Intrusion Detection System", Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS), pp. 21-26, 2016

[10] S. Zhai, Y. Cheng, W. Lu, and Z. Zhang, "Deep Structured Energy Based Models for Anomaly Detection", International Conference on Machine Learning, pp 1100-1109, 2016

[11] S. Aktar and A. Y. Nur, "Towards DDoS Attack Detection using Deep Learning Approach", Computers & Security, Vol. 129, 2023

[12] M. Sakurada and T. Yairi, "Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction", Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis, pp. 4-11, 2014

[13] I. Goodfellow, Y. Bengio, and A. Courville, "Deep Learning", MIT Press, 2016

[14] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive Auto-Encoders: Explicit Invariance During Feature Extraction", Proceedings of the 28th International Conference on Machine Learning (ICML-11), pp. 833-840, 2011