Anomaly detection in network logs

Andrei Cristian Airinei, Cătălin-Andrei Stan and Marius Drăgoi*
University of Bucharest, Romania
*Bitdefender, Romania

1. Introduction

Logging events is the primary method for recording the status of a computer network, thus, an automatic log anomaly detection system would be a valuable tool for monitoring the network. In this paper we proprose two solutions. First we treat this problem as classification task in a supervised fashion. Then, we address it by treating it as a masked language model and train it in a self-supervised manner.

2. Dataset and Preprocessing

Dataset Description:
We will be using the CSE-CIC-IDS2018 dataset [1] which includes a variety of simulated attacks such as Brute Force, DDoS, Web Attacks and so on. All these attacks will be treated as outliers.

Dataset stats	
Benign	83%
Attacks	17%

Data preprocessing: The timestamp should not affect whether a log is benign or not, so we will delete this column. The protocol and the destination port are equivalent, and since the port is already numerical, we will drop the protocol. Logs with negative values for certain features are considered noisy and will be deleted. Additionally, columns that contain only zero values will be dropped. Finally, we will remove columns where a significant portion of the dataset has negative values. After these adjustments, the dataset will have 68 numerical features.

5. Qualitative Analysis

We can observe that our method works exceptionally well for attacks that rely heavily on the network level (utilizing packets extensively), compared to attacks such as SQL Injection, Brute Force XSS, Infiltration, and Brute Force Web. These latter attacks involve user code content, classical exploitation of vulnerable software, and brute forcing user input, respectively.

7. References

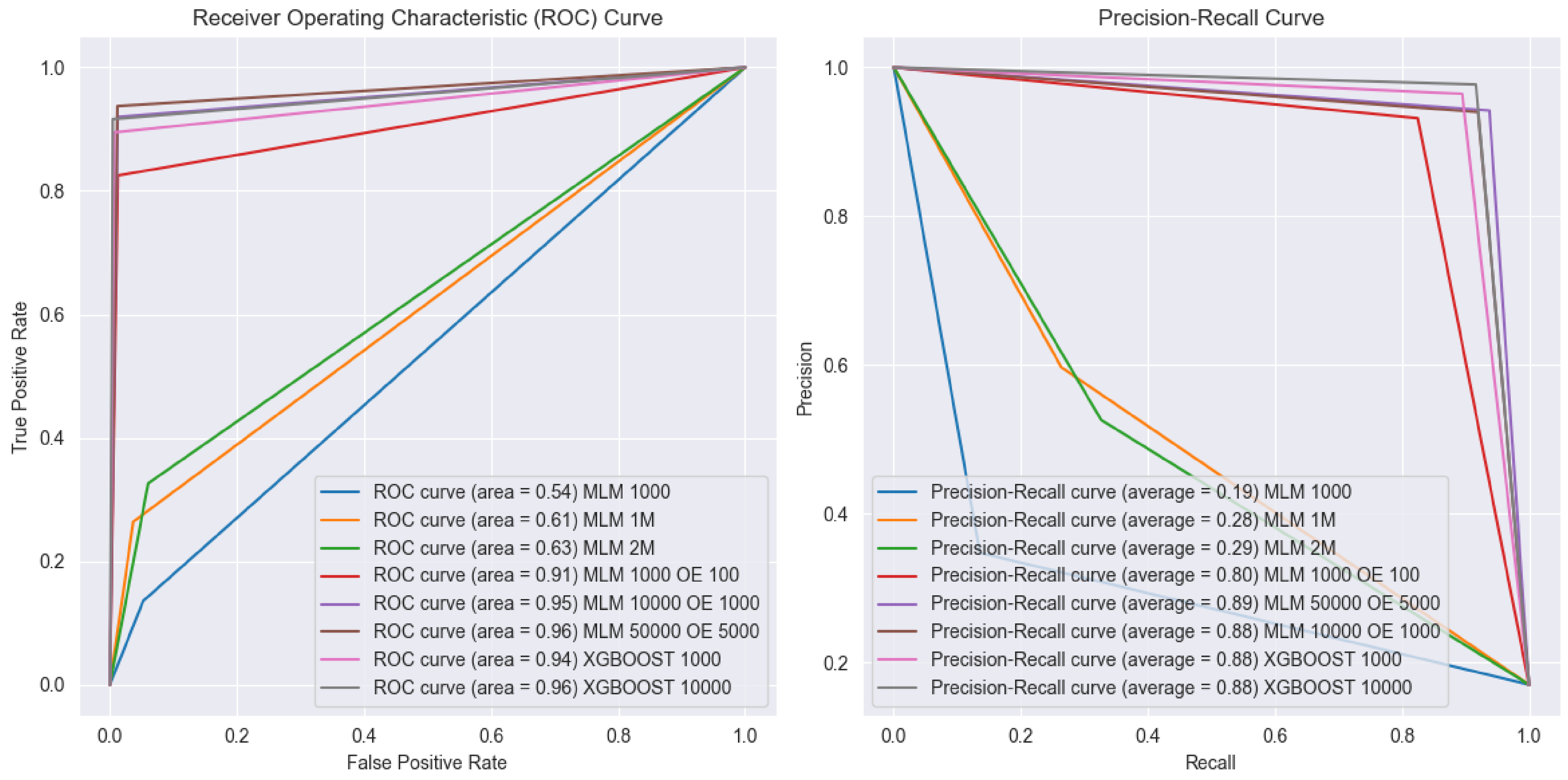
[1] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani. "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization". In: *International Conference on Information Systems Security and Privacy.* 2018. URL: `https://api.semanticscholar.org/CorpusID:4707749`.

3. Models

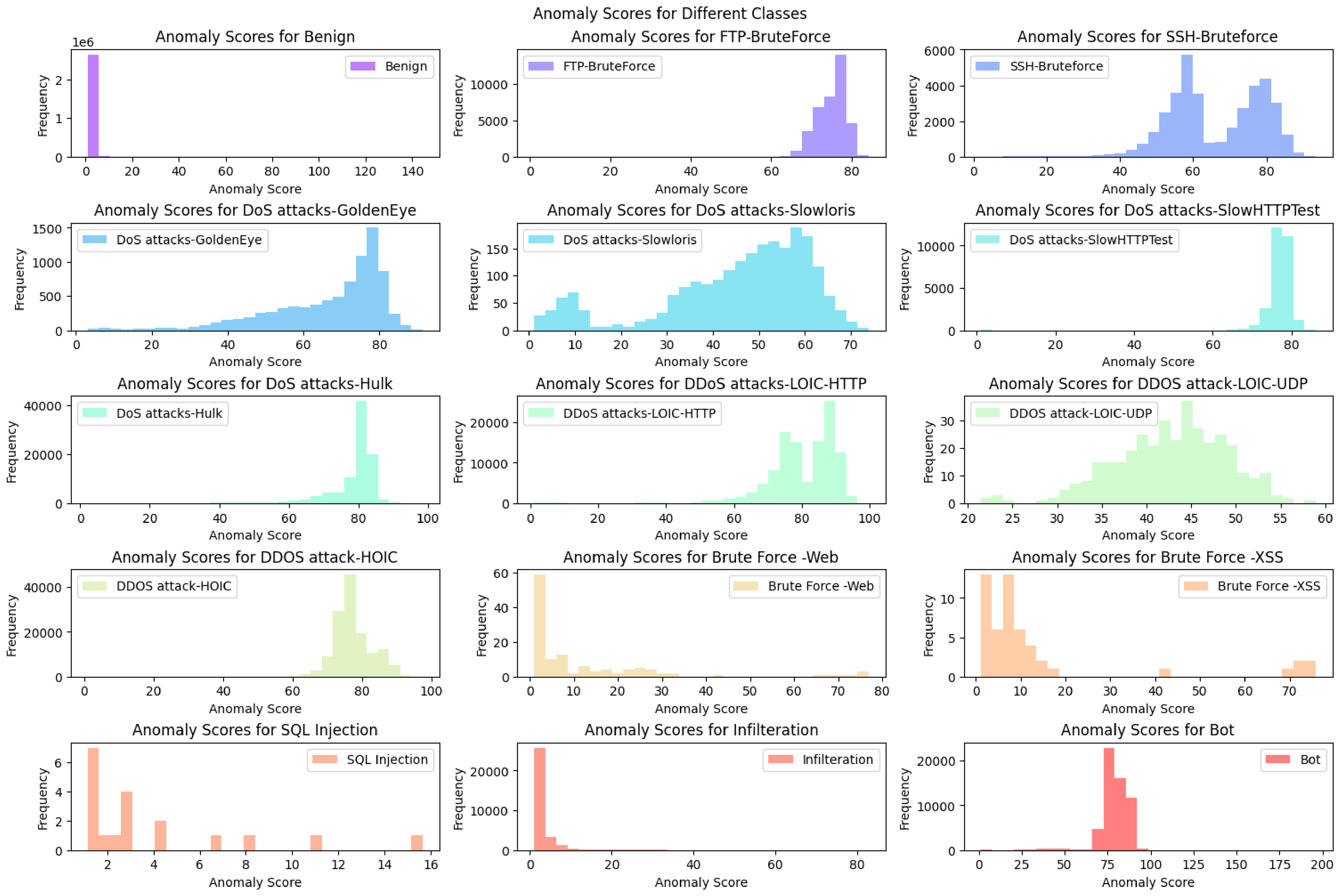
Supervised Classification: For the classification task, we split the dataset into a training set (80%) and a testing set (20%) and then train an XGBoost classifier. The main hyperparameters we searched for were the number of trees, the maximum depth of each tree, and the learning rate
Unsupervised: For this task, we split the dataset into training (60%), validation (20%), and testing (20%) sets. During training, we used only benign traffic and masked 15% of the log features. We then trained a BERT model to predict these masked tokens. For log classification, we also masked 15% of the features and calculated the anomaly score as the mean of the inverse probability of the most likely token for each mask. The validation set was used to determine the best threshold, and we then evaluated the model on the testing set. Additionally, we employed Outlier Exposure by adding a small portion of outliers and updating the loss function with the KL divergence between the softmax distribution for each masked token in the outliers and the uniform distribution. We binned each feature using the rank percentile and used 0 as the mask token. The model was configured with a vocabulary size of 102, a hidden size of 256, 2 hidden layers, 2 attention heads, and an intermediate size of 256.

4. Results

Below are the results of the proposed models, varying the training set size.



Here we present the anomaly scores for different types of attacks obtained using the second approach with outlier exposure, trained on the entire set of benign logs and 10,000 outliers. This approach achieved a 0.96 ROC AUC score.



6. Conclusion

We approached the anomaly detection task from both supervised and unsupervised perspectives. Additionally, we examined how dataset size impacts model performance. Our findings indicate that in data-scarce environments, using unsupervised techniques can yield good results.