MAUN Fen Bil. Dergi., 13, 2, 341-350 Araştırma Makalesi/ Research Article
MAUN J. of Sci., 13, 2, 341-350 DOI : 10.18586/msufbd.1753107

AutoGluon-Based Performance Analysis for Multi-Class Network Attack Detection

Sinan KOCAGÖZ[1], Fatih YÜCALAR[1]✉, Emin BORANDAĞ[1], Ender ŞAHİNASLAN[2]
[1] Department Of Software Engineering, Hasan Ferdi Turgutlu Faculty Of Technology, Manisa Celal Bayar University, Manisa, Türkiye
[2] Management Informatıon Systems Programme, Faculty Of Arts And Social Sciences, Mudanya University, Bursa, Türkiye

✉: fatih.yucalar@cbu.edu.tr [1] 0009-0005-8298-5134 [2] 0000-0002-1006-2227
[3] 0000-0001-5553-2707 [4] 0000-0001-8519-7612

Geliş (Received): 29.07.2025 Düzeltme (Revision): 09.09.2025 Kabul (Accepted): 11.09.2025

ABSTRACT

The increasing complexity and critical nature of cyber threats have heightened the importance of effective attack detection systems. In this study, various machine learning algorithms (SVM, KNN, Logistic Regression, Random Forest, XGBoost), deep learning models (CNN, LSTM, DNN), and the AutoML-based AutoGluon framework are systematically compared for multi-class network attack detection. The experiments utilize the UNSW-NB15 dataset. Due to the imbalanced class distribution in the dataset, class balancing was applied in certain analyses using the SMOTE technique. All models were evaluated using commonly adopted classification metrics, including Accuracy, Precision, Recall, F1-score, and ROC-AUC. The findings indicate that AutoGluon achieved the highest performance, owing to its automated modeling and ensemble-based approach. These results suggest that automated modeling techniques may offer greater competitiveness and effectiveness compared to traditional methods. By systematically analyzing the performance of different modeling strategies in intrusion detection systems, this study aims to provide guidance for the development of future security solutions.

Keywords: AutoGluon, AutoML, Deep Learning, Machine Learning, Network Attack Detection.

Çok Sınıflı Ağ Saldırısı Tespiti için AutoGluon Tabanlı Performans Analizi

ÖZ

Siber tehditlerin artan karmaşıklığı ve kritik doğası, etkili saldırı tespit sistemlerinin önemini artırmıştır. Bu çalışmada, çok sınıflı ağ saldırısı tespiti için çeşitli makine öğrenmesi algoritmaları (SVM, KNN, Lojistik Regresyon, Rastgele Orman, XGBoost), derin öğrenme modelleri (CNN, LSTM, DNN) ve AutoML tabanlı AutoGluon çerçevesi sistematik olarak karşılaştırılmıştır. Deneylerde UNSW-NB15 veri seti kullanılmıştır. Veri setindeki dengesiz sınıf dağılımı nedeniyle, bazı analizlerde SMOTE tekniği kullanılarak sınıf dengesi sağlanmıştır. Tüm modeller, Doğruluk, Kesinlik, Duyarlılık, F1-skoru ve ROC-AUC gibi yaygın sınıflandırma ölçütleri kullanılarak değerlendirilmiştir. Elde edilen bulgular, AutoGluon'un otomatik modelleme ve topluluk (ensemble) tabanlı yaklaşımı sayesinde en yüksek performansı sağladığını ortaya koymuştur. Bu sonuçlar, otomatik modelleme tekniklerinin geleneksel yöntemlere kıyasla daha rekabetçi ve etkili olabileceğini göstermektedir. Çalışma, saldırı tespit sistemlerinde farklı modelleme stratejilerinin performansını sistematik olarak analiz ederek, gelecekteki güvenlik çözümlerinin geliştirilmesine yönelik yol gösterici olmayı amaçlamaktadır.

Anahtar Kelimeler: AutoGluon, AutoML, Derin Öğrenme, Makine Öğrenmesi, Ağ Saldırısı Tespiti

INTRODUCTION

With the widespread adoption of the internet, dependency on digital systems has increased rapidly, rendering network infrastructures more vulnerable to malicious attacks. In particular, the security of network systems used in critical domains such as finance, healthcare, and defense must be ensured through proactive measures against such threats. Intrusion Detection Systems (IDS) are crucial for safeguarding system security, as they identify anomalous network traffic activities. Traditional signature-based methods have limitations, prompting researchers to explore machine learning (ML) and deep learning (DL)-based

approaches. These techniques significantly boost IDS accuracy and effectiveness by automatically classifying diverse attack types. However, zero-day attacks and class imbalance issues in datasets can hinder the detection of minority classes, thereby negatively impacting model performance. Therefore, this study investigates the impact of class imbalance on detection performance under various scenarios. For this purpose, the UNSW-NB15 dataset [1] was utilized to detect multi-class network attacks, and a comparative evaluation was conducted involving traditional ML algorithms (Logistic Regression, Support Vector Machine, K-Nearest Neighbors, Random Forest, Extreme Gradient Boosting), DL architectures (Dense Neural Network, Convolutional Neural Network, Long Short-Term Memory), and the AutoML-based AutoGluon framework. To address class imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) was applied in selected scenarios. This allowed for a comparative analysis of model performance using both balanced and imbalanced versions of the dataset [2, 3]. SMOTE was not employed in all experiments but only in cases where a significant impact on performance was anticipated. We evaluated all models' performance using key classification metrics: Accuracy, Precision, Recall, F1-score, and ROC-AUC. The paper's structure is as follows: Section 2 covers related work; Section 3 details materials and methods; Section 4 outlines experimental studies; and Section 5 discusses findings and conclusions.

RELATED WORK

Intrusion Detection Systems (IDS), vital for network security, are increasingly developed with ML and DL algorithms to combat rising cyber threats. In the literature, many studies employing datasets such as UNSW-NB15 have aimed to classify attacks using techniques like SVM, Random Forest, KNN, Naive Bayes, CNN, and LSTM, comparing their success rates. These studies typically involved manual processes such as model selection, hyperparameter tuning, and feature selection. However, a review of the literature reveals that AutoML-based approaches—which provide automatic model selection and optimization—have not been sufficiently utilized in the field of intrusion detection, especially powerful frameworks like AutoGluon. Addressing this gap, our study aims to contribute to the literature by employing the AutoGluon platform to automatically train and compare various ML and DL models. Moreover, the study proposes novel hybrid structures that combine AutoGluon with strong models (e.g., Random Forest, DNN), yielding notable results in terms of both accuracy and computational efficiency. This work thus presents alternative and effective modeling approaches for intrusion detection.

Türkyılmaz and Şentürk (2021) investigated the impact of different ML algorithms on intrusion detection performance using the UNSW-NB15 dataset. The study utilized the Orange toolkit to compare models in terms of accuracy and efficiency and evaluated the effect of feature selection on algorithm performance [4].

Similarly, Şimşek and Atılgan (2021) assessed various ML algorithms for detecting DoS and DDoS attacks in Internet of Things (IoT) environments. Metrics such as accuracy, precision, recall, and F1-score were used to compare algorithms including Random Forest, AdaBoost, Decision Trees, Naive Bayes, Logistic Regression, SVM, and k-NN. Their work was primarily based on the KDDCup99 dataset and offered ML-based solutions tailored for IoT systems [5].

Kurt Pehlivanoğlu et al. (2023) combined the "Bot_IoT" and "ToN_IoT" datasets to form a new dataset for classifying DoS, DDoS, and scanning attacks. Among various ML algorithms tested, the Gradient Boosting algorithm achieved the highest performance in multi-class classification problems [6].

Ata and Kadhim (2018) evaluated ML-based IDS structures for detecting network attacks, discussing dataset features and limitations in detail, and analyzing the performance of different ML algorithms (especially ANN) across various datasets. The study highlighted that Artificial Neural Networks yielded the best performance on average, emphasizing the potential of ML-based IDS systems [7].

Amarouche and Küçük (2022) compared the performance of ML and DL-based models in detecting network attacks using the UNSW-NB15 dataset. By measuring feature importance via the Random Forest algorithm, an optimal feature set was created, and models were tested in both multi-class and binary classification scenarios. While neural networks automatically learned critical features, ML models benefited from manual feature selection. The results indicated that DL architectures provided better generalization with lower false positive rates [8].

Çimen et al. (2021) conducted binary classification (normal vs. attack) on the NSL-KDD dataset using SVM, KNN, BayesNet, Naive Bayes, J48, and Random Forest algorithms. Classification was carried out using the WEKA software, and KNN achieved the highest accuracy rate of 98.12%. The study also assessed the impact of the number of neighbors and folds on results and showed that feature selection enhanced model performance [9].

Keskin & Okatan (2023) performed a multi-class intrusion detection study on the CSE-CIC-IDS2018 dataset using Decision Trees, Random Forest, Extra Trees, and Extreme Gradient Boosting algorithms. The dataset includes various attack types such as Brute Force, DDoS, Infiltration, Web, and Botnet. Recursive Feature Elimination (RFE) was used for feature selection, and XGBoost achieved the highest accuracy of 98.18% [10].

Erçin and Yolaçan (2021) proposed a comprehensive preprocessing and feature extraction method on a custom dataset consisting of 13,157 malicious and 10,000 normal HTTP requests to detect SQLi and XSS attacks. The study compared DL models such as CNN, LSTM, MLP with ML algorithms like SVM, KNN, and RF. Symbolization and Euclidean distance-based features were employed to improve detection accuracy, with a

strong emphasis on achieving low false alarm rates and high precision [11].

Biçakçı and Toklu (2022) proposed a multi-class intrusion detection system using traditional ML algorithms and a hybrid feature reduction method based on SAE and SelectKBest on the NSL-KDD dataset. Their SAE-SKB-RF model achieved an accuracy rate of 98.67%. However, DL and AutoML-based approaches were not explored in this study [12].

Finally, Koyuncu and Ünlü (2022) examined the development process of AI-based IDS, highlighting the limitations of traditional methods and emphasizing the effectiveness of ML, expert systems, and artificial neural networks in IDS design. Nonetheless, the study did not address AutoML-based systems [13].

MATERIALS AND METHODS

This section presents the details of the dataset used in the study, data preprocessing steps, class imbalance and SMOTE application, the algorithms and models employed, and the evaluation metrics.

Dataset

In this study, the UNSW-NB15 dataset was utilized to detect network attacks in a multi-class setting. Designed to resemble real network traffic, this dataset is well-suited for experimental comparisons due to its multi-class structure, which includes both various types of attacks and normal traffic. It contains 45 primary features and 10 distinct classes (labels) representing normal and anomalous (malicious) network behaviors. One of these classes corresponds to "Normal" traffic, while the remaining nine represent different categories of attacks. In the data preprocessing phase, One-Hot Encoding transformed categorical variables into numerical form, increasing the total feature count to 198. Table 1 summarizes the UNSW-NB15 dataset's fundamental characteristics.

Table 1. Fundamental characteristics of the UNSW-NB15 dataset

Feature	Value
Total Number of Samples	257,673
Number of Classes	8 (7 attack + 1 normal)
Number of Features	198
Training/Test Ratio	%80 / %20

Thanks to these properties, UNSW-NB15 is widely used as a benchmark dataset for evaluating the performance of ML and DL models developed for multi-class network intrusion detection. Nonetheless, the extremely low sample counts observed in certain attack categories were deemed likely to impair model training and exacerbate the class imbalance issue. To mitigate this, two attack types with instance counts below a predefined threshold were excluded from the experimental setup. Following this exclusion, a total of eight classes (seven attack types and one normal traffic class) were retained for training and evaluation processes. The retained attack categories within the UNSW-NB15 dataset are summarized in Table 2.

Table 2. Attack types in the UNSW-NB15 dataset

Attack Type	Description
Fuzzing	Test attacks aimed at detecting software vulnerabilities
Backdoors	Unauthorized access to systems via backdoors
Analysis	Passive attacks using network analysis tools
Exploits	Active attacks exploiting system vulnerabilities
Generic	Encryption-based attack types
DoS	Denial-of-service attacks
Reconnaissance	Information gathering and scanning activities
Normal	Normal traffic without any attack

Data Preprocessing

In order to ensure the reliable operation of the model, several data preprocessing steps were applied to the UNSW-NB15 dataset. These steps included handling missing values, data transformation, feature scaling, label encoding, and the partitioning of the dataset into training and testing subsets. All of these processes are essential for improving model performance and maintaining the integrity of the input data.

Data Cleaning

The dataset was initially examined for missing values. Records containing any incomplete entries were removed to ensure data integrity. Additionally, columns with constant values—which do not contribute meaningful information to the analysis—were eliminated. Duplicate records were also identified and excluded to avoid redundancy and reduce potential bias during the model training process.

Class Selection and Filtering

In the initial phase, the model was trained using all attack classes in the dataset. However, the presence of certain attack types with a very low number of instances led to class imbalance, which negatively impacted the overall model performance. To mitigate this issue, classes with inadequate sample sizes were removed from the analysis. Consequently, the training and evaluation processes were carried out using a total of eight classes—seven attack types and one normal traffic class.

Feature Transformation

Categorical features (e.g., proto, service, and state) were transformed into numerical format using one-hot encoding to make them compatible with the model's learning process. Additionally, the target variable, consisting of class labels, was converted into numerical values using label encoding to ensure compatibility with classification algorithms.

Class Imbalance and SMOTE Application

There is a significant class imbalance in the dataset; while normal traffic samples are abundant, the number of examples for certain attack types is considerably low. This imbalance causes models to predominantly predict the majority class, "normal," leading to poor detection of attacks. To address this issue, we applied for the Synthetic Minority Oversampling Technique (SMOTE) to generate synthetic samples for the minority classes. SMOTE aims to improve the model's ability to learn these underrepresented classes by oversampling existing attack instances. Although initially promising, the application of SMOTE did not yield the desired overall performance improvement. While partial increases in recall and F1-scores were observed for some attack types, the overall accuracy decreased, and models trained on synthetic data produced misleading results on real attacks. SMOTE was tested exclusively with ML algorithms and was not applied to DL models due to the unsatisfactory results obtained.

Algorithms and Models Used

In this study, we conducted a comparative analysis of different modeling approaches for the detection of multi-class network attacks. The models were evaluated under three main categories:

- Traditional ML algorithms (classical methods such as Random Forest, SVM, and XGBoost),
- DL architectures (ANN and CNN),
- The AutoGluon platform for automated model selection based on AutoML.

Furthermore, beyond using AutoGluon alone, we developed hybrid models by combining it with other powerful algorithms. For instance, combinations of AutoGluon with XGBoost and AutoGluon with CNN were created, and the performance of these hybrid models was compared against that of the standalone models. This approach provided a significant contribution by assessing the potential to enhance model performance.

Machine Learning Algorithms

In this study, fundamental ML algorithms known for their distinct features and advantages were comparatively evaluated for multi-class network attack detection.

Logistic Regression (LR), a linear classification technique, models the relationship between variables using a sigmoid function to estimate probabilities. It delivers fast and effective results when the data is linearly separable. Additionally, the provision of class probabilities enhances interpretability, which is why it was selected as a baseline model in this study [14].

Support Vector Machine (SVM) is designed to identify the optimal hyperplane that most effectively separates the classes. Particularly effective on high-dimensional and noise-free datasets, this algorithm constructs decision boundaries by focusing on support vectors. Consequently, it can achieve strong classification performance even with limited sample sizes. Within this study, it was tested as a robust baseline model for multi-class attack detection [15].

The *K-Nearest Neighbors (KNN)* algorithm classifies a data point based on the majority class among its nearest neighbors. It performs well on datasets with well-distributed data but becomes computationally expensive as dataset size increases. If distinct patterns exist among attack types, KNN can yield accurate classification. Therefore, it was included as a comparative model in this work [16].

Random Forest (RF) consists of an ensemble of decision trees, each trained on different data subsets and feature combinations, which effectively reduces the risk of overfitting. Its compatibility with both categorical and numerical variables makes it a reliable and robust model, especially suitable for complex data structures such as network traffic [17].

Extreme Gradient Boosting (XGBoost) is a boosting-based algorithm renowned for its high accuracy and error minimization capabilities. It sequentially trains decision trees, with each tree focusing on reducing the errors of previous models. It offers advanced hyperparameter optimization and computational efficiency. Frequently preferred for complex classification problems, XGBoost was among the top-performing models tested in this study [18].

Deep Learning Algorithms

In this study, different DL architectures were compared for multi-class network attack detection.

Dense Neural Network (DNN) is a fundamental artificial neural network model consisting of fully connected layers, where all neurons in each layer are interconnected. The model effectively learns nonlinear data relationships, showing strong performance in multi-class network traffic classification [19].

Although *Convolutional Neural Network (CNN)* is commonly used in image processing, in this study, it was applied to two-dimensional numerical data matrices. This architecture, composed of convolutional, pooling, and fully connected layers, achieved high success in distinguishing characteristic features of attack types [20].

Long Short-Term Memory (LSTM) network, developed to overcome the forgetting problem of classical RNNs, has the ability to learn patterns in temporal and sequential data. In this study, LSTM was used to model attack scenarios involving timing and sequence information in network traffic, yielding effective results [21].

Automatic Machine Learning: AutoGluon

Automatic Machine Learning (AutoML) is a comprehensive approach that automates processes such as model building, algorithm selection, and hyperparameter optimization in ML without requiring user intervention. In this study, the AutoGluon platform, developed by Amazon, was utilized. AutoGluon provides an end-to-end automated modeling pipeline, particularly effective when working with structured (tabular) data. It only requires the user to specify the dataset and the target label column; all other processes—including data

preprocessing, model selection, hyperparameter tuning, and ensemble modeling—are handled automatically by the system. Within this scope, the "TabularPredictor" component of AutoGluon was employed. This component constructs the final prediction model by combining sub-models trained with various algorithms, using stacking and bagging techniques. Despite its ease of use, AutoGluon demonstrates the potential to outperform many traditional ML methods in terms of prediction accuracy [22].

Hybrid Approaches: Model + AutoGluon

In this study, not only the models automatically generated by AutoGluon but also hybrid models that combine AutoGluon with several classical and DL algorithms were evaluated. The main objective of this approach is to enhance model performance by integrating high-performing individual models with AutoGluon's automatic optimization and ensemble learning infrastructure. The tested hybrid models are as follows:

- Random Forest + AutoGluon
- XGBoost + AutoGluon
- DNN + AutoGluon

These models were directly integrated into the AutoGluon system and included in the processes of hyperparameter optimization and model ensembling. Experimental results showed that, in some scenarios, these hybrid models achieved results comparable to or very close to those produced by AutoGluon's native models, while in other cases, they performed slightly worse than standalone AutoGluon models. AutoGluon is an AutoML platform developed for structured (tabular) data, and its modeling process is primarily based on ensemble learning techniques such as bagging and stacking. Bagging reduces variance by training models on diverse data subsets, while stacking boosts generalization by using a meta-model to combine outputs from multiple base models. In the hybrid modeling approach, strong individual models such as XGBoost, Random Forest, and DNN were directly integrated into AutoGluon's stacking framework. In doing so, the advantages of both classical and automated models were unified within a single ensemble, enabling more effective learning of complex attack patterns. Thanks to its flexible and robust architecture, AutoGluon produced competitive results with high accuracy and low false positive rates, even under challenging data conditions such as class imbalance.

Model Evaluation Metrics

To assess model effectiveness in multi-class network attack detection, we used common classification metrics like the confusion matrix, accuracy, precision, recall, and F1-score.

Confusion Matrix

The confusion matrix is a highly effective evaluation tool, tabulating a classification model's correct and incorrect predictions for each class. Since the UNSW-NB15 dataset used in this study has a multi-class structure, a square-shaped confusion matrix of size n x n was employed. In this matrix, rows represent the actual classes, and columns correspond to the model's predicted classes. Each cell indicates the frequency with which instances of a particular actual class were predicted as each possible class by the model. The resulting distribution of predictions across all classes is illustrated in Table 3 [23].

Table 3. Confusion matrix

	Predicted Positive	**Predicted Negative**
Actual Positive	TP	FN
Actual Negative	FP	TN

Table 3 provides the definitions of the four fundamental components used in evaluating classification model performance. A True Positive (TP) occurs when the model accurately identifies an instance as belonging to a particular class, matching the true class label. A False Positive (FP) happens when the model wrongly assigns an instance to a class it does not actually belong to. Conversely, a False Negative (FN) indicates that the model fails to identify an instance as belonging to its true class and instead assigns it to a different class. Lastly, a True Negative (TN) represents the scenario in which the model correctly identifies that an instance does not belong to a given class and thus assigns it elsewhere appropriately.

Accuracy

Accuracy is defined as the proportion of correctly classified instances relative to the total number of instances in the dataset. The formula used to compute accuracy is presented in Equation (1).

$$Accuracy = \frac{\sum_{i=1}^{n} TP_i}{TP + TN + FP + FN} \quad (1)$$

In Equation (1), *TPi* denotes the number of correctly predicted instances for class *i*, while *n* represents the total number of classes in the dataset [23].

Precision

Precision evaluates the ratio of true positive predictions to the total instances predicted as positive by the model for a specific class. The formula used to calculate precision is presented in Equation (2) [23].

$$Precision_i = \frac{TP_i}{TP_i + FP_i} \quad (2)$$

Recall

Recall measures the proportion of true positive instances that the model correctly identifies. The formula used to calculate recall is provided in Equation (3) [23].

$$Recall_i = \frac{TP_i}{TP_i + FN_i} \quad (3)$$

F1-score

F1-score represents the harmonic meaning of Precision and Recall, offering a balanced metric that accounts for both false positives and false negatives. The formula used to calculate the F1-score is presented in Equation (4) [23].

$$F1 - score_i = 2 \times \frac{Precision_i \times Recall_i}{Precision_i + Recall_i} \quad (4)$$

EXPERIMENTAL STUDIES AND RESULTS

In this section, we describe the modeling process conducted using the UNSW-NB15 dataset. Various model types were tested to determine which performs most effectively in detecting network attacks.

Data Preparation and Splitting

The dataset was split into training and testing subsets, allocating 80% for training and 20% for testing. Although the original UNSW-NB15 dataset contains a total of 10 classes (1 normal and 9 attack types), two attack classes with sample sizes below a certain threshold (e.g., Worms and Shellcode) were excluded from the analysis. Consequently, modeling was performed using the remaining 8 classes (7 attack types and 1 normal class). This class selection process was applied exclusively to the ML, DL, and hybrid models; the AutoGluon platform conducted automatic model training using the complete class structure without exclusion.

SMOTE-Based Data Balancing Approach

SMOTE was applied to investigate the impact of class imbalance on ML algorithms. The method aims to balance class distributions by generating synthetic samples for the minority class through interpolation among its instances. Since SMOTE was originally developed for structured data and operates based on proximity and/or vector interpolation, it is not commonly applied to DL models [24]. Therefore, in this study, SMOTE was utilized exclusively with traditional ML algorithms, while DL, AutoGluon, and hybrid models were excluded. This approach enabled a comparative analysis of the effects of class imbalance across different modeling scenarios and was intended to enhance the performance of ML-based models.

Tested Models

In this study, three primary modeling approaches were employed:

- ML Algorithms: LR, SVM, KNN, RF, and XGBoost were evaluated. Each algorithm was tested in both SMOTE-applied and non-SMOTE versions to assess the impact of class imbalance.
- DL Architectures: Three distinct neural network structures were selected: DNN, CNN, and LSTM. The layer configurations, activation functions, and training parameters were kept consistent across models to ensure a fair comparison.
- AutoML Approach: Amazon's AutoGluon platform was utilized, which autonomously performs model selection, hyperparameter tuning, and ensemble model construction without requiring user intervention.

Hybrid Model Architectures

In addition to evaluating each approach individually, we also developed hybrid models by combining AutoGluon with other algorithms. The aim was to leverage the strengths of both conventional and automated learning methods. Within this scope, the following hybrid configurations were tested:

- Random Forest + AutoGluon
- XGBoost + AutoGluon
- DNN + AutoGluon

Evaluation Process

All models were evaluated using standard classification metrics such as Accuracy, Precision, Recall, F1-score, and ROC-AUC. Furthermore, confusion matrices were generated for each model to provide a detailed analysis of misclassification patterns at the class level. The parameter settings and layer configurations used in the implementation of ML and DL algorithms, as well as ensemble models, are also described. The parameter values for ML algorithms are shown in Table 4.

Table 4. Parameter settings of ML algorithms

Algorithm	Parameter	Value
Logistic Regression	Solver	liblinear
	Max Iter	1000
	C	1.0
	Random State	42
Random Forest	n_estimators	100
	Max Depth	10
	Random State	42
XGBoost	n_estimators	100
	learning_rate	0.1
	eval_metric	logloss
	random_state	42
KNN	n_neighbors	5
SVM	Kernel	rbf
	C	1.0
	Gamma	scale
	probability	True

The parameters and layer values used in executing the DL algorithms are presented in Table 5.

Table 5. Parameter settings of DL algorithms

Algorithm	Parameter	Value
DNN	Layers	{512, 256, 128, 64}
	Activation	ReLU
	Dropout	0.4, 0.4, 0.3, 0.2
	Optimizer	Adam (learning_rate=0.001)
	Epochs	20
	Batch Size	128
	Validation Split	0.2
	Regularization	L1=1e^{-5}, L2=1e^{-4}
LSTM	Layers	{128,64,128,64}
	Dropout	0.3
	Optimizer	Adam (learning_rate=0.001)
	Epochs	20
	Activation	default
	Batch Size	128
	Validation Split	0.2
CNN	Convolution Layer	Conv1D (filters=64, kernel_size=3, activation=relu)
	Pooling Layer	MaxPooling1D (pool_size=2)
	Dropout	0.5
	Dense Layer (Output)	Softmax
	Optimizer	Adam (learning_rate=0.001)
	Epochs	20
	Batch Size	128
	Validation Split	0.2

The algorithm parameter information used in the proposed ensemble model is provided in Table-6. For the use of AutoGluon, the default parameter values were selected.

Table 6. Parameter settings of ensemble model

Ensemble Model	Algorithm	Parameter	Value
RandomForest + AutoGluon	RandomForest	n_estimators	100
		random_state	42
XGBoost + AutoGluon	XGBoost	eval_metric	mlogloss
		random_state	42
DNN + AutoGluon	DNN	Layers	{128, 64, 32}
		Activation	relu
		Dense Layer	softmax
		Optimizer	Adam
		Epochs	10
		Batch Size	32

Experimental Results

In this section, the performance outcomes of different modeling approaches applied to the UNSW-NB15 dataset are presented and comparatively assessed. Common classification metrics, including Accuracy, Precision, Recall, F1-score, and ROC-AUC, have been utilized to evaluate the models' effectiveness. Table 7 provides a summary of the results obtained.

Table 7. Performance Results of ML, ML_Smote, DL, and Ensemble Models on the UNSW-NB15 Dataset

Type	Algorithm	Accuracy	Precision	Recall	F1-score
ML	LR	0.76	0.74	0.76	0.73
	RF	0.81	0.83	0.81	0.77
	XGBoost	0.83	0.82	0.82	0.80
	KNN	0.77	0.76	0.76	0.76
Average		0.79	0.79	0.79	0.76
ML-Smote	LR	0.69	0.81	0.69	0.72
	RF	0.75	0.84	0.75	0.77
	XGBoost	0.80	0.84	0.79	0.81
	KNN	0.74	0.77	0.74	0.75
	SVM	0.75	0.83	0.73	0.75
Average		0.74	0.82	0.74	0.76
DL	LSTM	0.51	0.23	0.28	0.24
	CNN	0.80	0.70	0.68	0.68
	DNN	0.82	0.73	0.67	0.66
Average		0.71	0.55	0.54	0.53
Ensemble	RF + AutoGluon	0.85	0.73	0.61	0.62
	XGBoost + AutoGluon	0.85	0.73	0.61	0.62
	DNN + AutoGluon	0.84	0.73	0.60	0.61
Average		**0.85**	**0.73**	**0.61**	**0.62**

According to the results presented in Table 7, traditional ML algorithms generally produced accuracy scores ranging from 76% to 83%. Among these algorithms, XGBoost emerged as the most successful model, achieving an accuracy of 83% and a ROC-AUC score of 96.49%. In scenarios where SMOTE was applied, a notable improvement was observed in the recall and F1-score values, particularly due to the enhanced representation of minority classes. However, a slight decrease in accuracy was observed in some models (e.g., Logistic Regression). This suggests that although SMOTE may have a limited impact on overall accuracy, it significantly contributes to improving class balance. According to the results presented in Table 7, among the DL models, the DNN achieved the highest performance with an accuracy of 82.18% and a ROC-AUC score of 97.41%.

In contrast, the LSTM model, despite its architecture designed for temporal dependencies, demonstrated unexpectedly low performance, yielding only 51.38% accuracy. This outcome may be attributed to the absence of temporal features in the dataset, which could have limited the learning capability of the LSTM model. AutoML-based AutoGluon models, although evaluated without the application of SMOTE, produced satisfactory and competitive results with an average accuracy of 84.66%. This level of accuracy is quite promising when compared with both traditional ML and DL models. However, it is noteworthy that AutoGluon models exhibited relatively lower performance in class-imbalance-sensitive metrics such as precision and recall. The AUC values obtained for the ensemble models are presented in Table 8.

Table 8. AUC values for ensemble models

Type	Algorithm	AUC
Ensemble	RF + AutoGluon	0.9883
	XGBoost + AutoGluon	0.9897
	DNN + AutoGluon	0.9891

Figure 1 presents the confusion matrices for the hybrid models RF+AutoGluon, XGBoost+AutoGluon, and DNN+AutoGluon, providing a visual representation of each model's class-wise prediction performance in greater detail. The ROC curves of these ensemble models are illustrated in Figure 2.

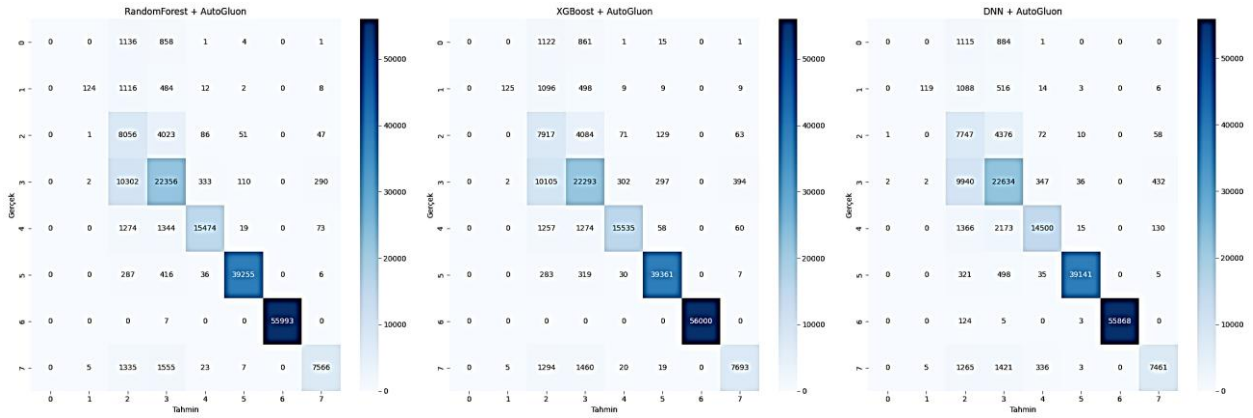

Figure 1. Confusion matrices resulting from the application of RandomForest, XGBoost, and DNN algorithms in conjunction with AutoGluon.

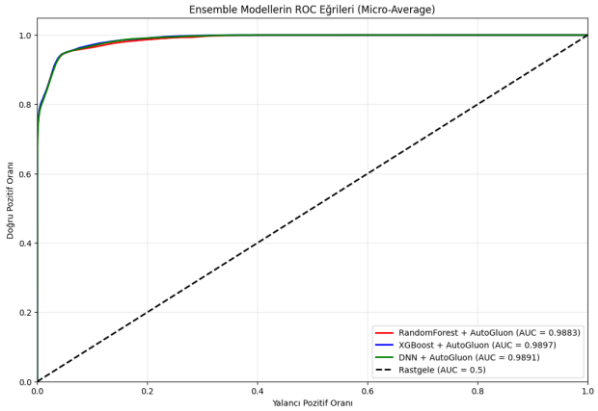

Figure 2. ROC curves of ensemble models

CONCLUSION AND FUTURE WORK

In this study, classical ML algorithms, DL models, and the AutoML-based AutoGluon framework were comparatively evaluated for multi-class attack detection using the UNSW-NB15 dataset. Experimental results demonstrated that AutoGluon and hybrid model combinations notably outperformed others in terms of overall accuracy and ROC-AUC performance.

In future studies, techniques like Bayesian optimization and grid search can be utilized to further improve model performance. Additionally, the scope of the study can be expanded by incorporating next-generation datasets like CIC-IDS2017 and CSE-CIC-IDS2018. From an application perspective, testing the developed system on real-time network traffic will be a critical step toward validating its practical effectiveness. Finally, creating a proprietary dataset derived entirely from real network traffic would enable a more precise evaluation of the attack detection models' efficacy in authentic scenarios.

REFERENCES

[1] Nour, M., Slay, J. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). *2015 Military Communications and Information Systems Conference (MilCIS)*, IEEE, 2015.

[2] Chawla, N. V., Bowyer, K. W., Hall, L. O., Kegelmeyer, W. P. SMOTE: Synthetic Minority Over-sampling Technique, *Journal of Artificial Intelligence Research*, 16, 321–357, 2002.

[3] Fernández, A., García, S., Herrera, F., Chawla, N. V. SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of Artificial Intelligence Research*. 61(1): 863–905, 2018.

[4] Türkyılmaz, Y., Şentürk, A. Saldırı tespitinde makine öğrenmesi yöntemlerinin performans analizi. *Avrupa Bilim ve Teknoloji Dergisi*, (32), 107–112, 2021.

[5] Şimşek, M. M., Atılgan, E. DoS and DDoS Attacks on Internet of Things and Their Detection by Machine Learning Algorithms. *European Journal of Science and Technology*, 32, 107–112, 2021.

[6] Kurt Pehlivanoğlu, M., Kuyucu, A., Kaya, R., & Aydın, R. IoT Veri Kümelerinde Makine Öğrenmesine Dayalı Saldırı Tespiti. *Avrupa Bilim ve Teknoloji Dergisi*, 52, 19–26, 2023.

[7] Ata, O., Kadhim, K. Network Intrusion Detection Using Machine Learning Techniques. *Aurum Journal of Engineering Systems and Architecture*, 2(1), 115–123, 2018.

[8] Amarouche, S., Küçük, K. *Machine and deep learning-based intrusion detection and comparison in Internet of Things*. Journal of Naval Sciences and Engineering, 18(2), 333–361, 2022.

[9] Çimen, F. M., Sönmez, Y., İlbaş, M. Performance analysis of machine learning algorithms in intrusion detection systems. *Düzce University Journal of Science and Technology*, 9, 251–258, 2021.

[10] Keskin, S., Okatan, E. Machine learning methods for intrusion detection in computer networks: A comparative analysis. *International Journal of Engineering and Innovative Research*, *5*(3), 268–279, 2023.

[11] Erçin, M. S., Yolaçan, E. N. SQLi ve XSS saldırı tespitinde kullanılan yeni bir özellik çıkarma yöntemi. *Uluslararası Bilgi Güvenliği Mühendisliği Dergisi*, *8*(1), 1–11, 2021.

[12] Biçakçı, M. S., Toklu, S. Bilgisayar ağı güvenliği için hibrit öznitelik azaltma ile makine öğrenmesine dayalı bir saldırı tespit sistemi tasarımı. *GBAD – Uluslararası Bilgi Güvenliği Mühendisliği Dergisi*, 11(3), 203–220, 2022.

[13] Koyuncu, M. D., Ünlü, N. Yapay zekâ tekniklerinin saldırı tespit sistemlerinde kullanımı. *Beykoz Akademi Dergisi*, 10(1), 78–87, 2022.

[14] Maalouf, M. Logistic regression in data analysis: an overview. *International Journal of Data Analysis Techniques and Strategies*, 3(3), 281-299, 2011.

[15] Pisner, D. A., Schnyer, D. M. Support vector machine. In *Machine learning*. Academic Press, pp. 101-121, 2020.

[16] Zhang, Z. Introduction to machine learning: k-nearest neighbors. *Annals of translational medicine*, 4(11), 218, 2016.

[17] Belgiu, M., & Drăguţ, L. Random Forest in remote sensing: A review of applications and future directions. *ISPRS journal of photogrammetry and remote sensing*, 114, 24-31, 2016.

[18] Osman, A. I. A., Ahmed, A. N., Chow, M. F., Huang, Y. F., El-Shafie, A. Extreme gradient boosting (Xgboost) model to predict the groundwater levels in Selangor Malaysia. *Ain Shams Engineering Journal*, *12*(2), 1545-1556, 2021.

[19] Pelt, D. M., & Sethian, J. A. A mixed-scale dense convolutional neural network for image analysis. *Proceedings of the National Academy of Sciences*, *115*(2), 254-259, 2018.

[20] Purwono, P., Ma'arif, A., Rahmaniar, W., Fathurrahman, H. I. K., Frisky, A. Z. K., & ul Haq, Q. M. Understanding of convolutional neural network (CNN): A review. *International Journal of Robotics and Control Systems*, *2*(4), 739-748, 2022.

[21] Graves, A., & Graves, A. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, 37-45. 2012.

[22] Erickson, N., Mueller, J., Shirkov, A., Zhang, H., Larroy, P., Li, M., Smola, A. Autogluon-tabular: Robust and accurate automl for structured data. *arXiv preprint arXiv:2003.06505*, 2020.

[23] Borandağ, E. LSRM: A New Method for Turkish Text Classification. *Appl. Sci*. 14, 11143, .2024.

[24] Cheng, W. C., Mai, T. H., Lin, H. T. From SMOTE to Mixup for Deep Imbalanced Classification. In: Lee, C. Y., Lin, C. L., Chang, H. T. (eds) Technologies and Applications of Artificial Intelligence. TAAI 2023. *Communications in Computer and Information Science*, vol 2074. Springer, Singapore, 2024.