Leveraging CNNs, Quantization, and Random Forest for Edge Deployable Intrusion Detection Efficiency

Anushika Kothari[1], Shreepad Joshi[1], Shreya Pai[1], Ritesh Hiremath[1], Priyadarshini Patil[1], and Dr. Meena S M[1]

KLE Technological University, Hubballi, India
kotharianushika01@gmail.com, shreepadjoshi161@gmail.com, paishreya2423@gmail.com,riteshhiremath6@gmail.com, priyadarshini.patil@kletech.ac.in,msm@kletech.ac.in

Abstract. This research introduces a novel Intrusion Detection System (IDS) for edge computing, blending Convolutional Neural Networks (CNN) and Random Forests for feature reduction, with a focus on anomaly detection. The use of quantization techniques on ensembled CNN models is a key aspect, ensuring the system remains lightweight while preserving detection accuracy. This approach is particularly effective in resource-limited settings, responding to the need for efficient cybersecurity in such environments. The ensemble model, after quantization, achieved an accuracy of 98.13% and attained a 50% reduction in size, maintaining robust detection capabilities even in constrained resource scenarios. The study's results indicate a remarkable achievement in balancing computational efficiency with high detection performance, marking a significant step in edge-deployable cybersecurity technologies.

Keywords: Intrusion Detection System (IDS) · Convolutional Neural Networks (CNN) · Quantization · Ensemble Learning · Random Forest · Cybersecurity · Edge Computing.

1 Introduction

The significance of cybersecurity in today's highly interconnected digital world cannot be underestimated. Security breaches persist as an ever-evolving threat, posing ongoing challenges in safeguarding the integrity of digital systems. Sophisticated defense mechanisms are needed to meet this challenge, and Machine Learning (ML)-enabled Intrusion Detection Systems (IDS) have become essential tools for addressing digital security [1]. To handle the complexities of modern cybersecurity threats, this research explores the convergence of Machine Learning (ML) and security in the field of intrusion detection systems.

Intrusion Detection System (IDS) research is always changing to keep up with the ever-changing cybersecurity threat landscape. Researchers in this area investigate a variety of approaches to improve detection methods' efficacy [2].

To improve the performance of IDS models, ongoing studies include novel algorithms, feature selection techniques, and ensemble approaches. The overall development in this field is additionally aided by exploring edge computing for real-time threat detection, emphasizing the importance of implementing interpretable AI technologies. Anticipating and mitigating new cyber threats is still the major objective, as it keeps IDS frameworks relevant and resilient in protecting digital environments.

For real-time threat detection at the network's forefront, edge device deployment of IDS is essential. IDS on edge devices [3] lowers latency by processing data locally, guaranteeing quick detection and reaction to possible intrusions. This decentralized strategy improves overall cybersecurity in the changing digital ecosystem and is in line with the edge computing trend. Given the importance of edge devices in the Internet of Things ecosystem and their ability to effectively reinforce the security of interconnected systems, it is critical to strategically place IDS on them.

This research focuses on the construction of an IDS model with a particular focus on anomaly-based detection using deep learning techniques. Anomaly-based intrusion detection systems (AIDS) [4], are characterized by their ability to analyze networks' typical behavior and identify any abnormalities that may be signs of unusual activity. AIDS can adapt and identify new sorts of intrusions, which makes it especially effective against zero-day attacks, in contrast to signature-based intrusion detection systems [5], which rely on predetermined attack patterns. By using deep learning methods, the suggested model aims to improve anomaly-based detection's accuracy and flexibility, adding to an IDS framework that can handle new cybersecurity issues.

We present a strategy for improving IDSs that uses a streamlined methodology. Random Forest (RF) feature reduction maximizes model efficiency, and data transformation improves the interpretability of datasets. The detection core is made up of Convolutional Neural Networks (CNN), both ensembled and standalone, which use network patterns to accurately identify threats. For optimal computing efficiency, quantization is done, facilitating the implementation on edge devices. This combination supports the accuracy and effectiveness of our proposed IDS.

2 Related Works

Traditional methods, ML, and DL techniques represent the spectrum. This section provides a concise overview, focusing on key contributions that have shaped the evolution of Intrusion Detection Systems (IDS) research.

Petros Toupas et al. [6] conducted a study emphasizing multi-class classification in an IDS using Deep Neural Networks (DNNs). Their proposed architecture unveils a sophisticated design, featuring a foundational input layer with 44 features derived from extensive feature engineering. This sets the stage for a sequence of eight hidden layers, intricately structured with node configurations progressively decreasing from 140 to 20, then expanding to 120 in the

final layer. The network culminates in the softmax output layer, which generates probabilities for 13 classes, providing a comprehensive foundation for accurate predictions. With an outstanding accuracy of 99.88%, their model surpasses existing methods, leveraging the CICIDS2017 dataset for training and assessment. The paper proposes potential directions for future research, such as conducting a thorough examination of feature reduction, expanding datasets, and investigating alternative neural network architectures.

The study presented by Li Yang et al. [7] showcases a unique architecture for an IDS designed to recognize cyberattacks in vehicular networks, encompassing both internal and external security threats. The proposed system combines optimized DCNNs, transfer learning, ensembling, and hyper-parameter optimization (HPO). Experiments conducted on benchmark datasets, including Car-Hacking and CICDS2017, reveal superior performance compared to existing techniques. The research showcases the framework's effectiveness by introducing a data transformation technique to enhance pattern identification. Performance metrics validate its success in addressing cyber risks in IoV systems, achieving high accuracy, precision, recall, and F1-scores (ranging from 96.3% to 100.0%). This study introduces a tailored and impactful approach, making a significant contribution to cybersecurity in the Internet of Vehicles (IoV).

The research conducted by Md. Al Mehedi Hasan et al. [8] delves into the significance of feature selection in improving the efficacy of intrusion detection systems (IDS). The work effectively tackles the difficulties presented by large and redundant data by skillfully utilizing Random Forest (RF), intending to lower processing times and increase detection rates. On the KDD'99 dataset [9], the suggested two-step strategy successfully finds and ranks features with greater variable relevance, demonstrating a noteworthy decrease in input features, processing time, and considerable improvement in classification accuracy. The focus on classifier performance with a smaller feature set (25 features) as opposed to the entire set (41 features) highlights how important feature selection is for enhancing IDS accuracy and efficiency.

Our research stems from a comprehensive literature review, revealing certain gaps and challenges within the realm of IDSs. Existing studies often prioritize high accuracy through intricate models, which, while effective, can hinder edge deployment efficiency and interpretability. Notably, there is a need for more exploration into diverse neural network architectures to better understand their impact on IDS performance. Additionally, challenges persist in developing flexible and robust feature selection techniques tailored to the dynamic nature of intrusion patterns. Our proposed methodology actively addresses these gaps by incorporating ensemble techniques, and quantization methods, and evaluating various neural network topologies, all while optimizing feature selection to enhance the overall adaptability and effectiveness of IDS frameworks.

3 Methodology

3.1 Proposed System Overview

Our intrusion detection process initiates with rigorous data preprocessing on the CICIDS2017 dataset. In alignment with insights from cybersecurity papers [7] [8], our framework seamlessly integrates methodologies culminating in the development of a comprehensive IDS. Leveraging Random Forest for effective feature elimination, the dataset goes through a data transformation process. This holistic approach is further extended to implement quantization, ensuring the efficiency and adaptability of our system for real-world cybersecurity challenges.

We incorporate advanced CNN models to extract diverse representations. Subsequently, our solution is strengthened through ensemble techniques like concatenation, probability averaging, and bagging. This strategic approach not only elevates prediction accuracy but also enhances resistance to emerging threats, providing a robust defense.

A standout feature of our framework is its adaptability for edge deployment. We extend this adaptability by making our ensemble models quantizable, ensuring compatibility with resource-constrained edge devices. This strategic move streamlines intrusion detection in environments with limited computational resources, positioning our framework as a comprehensive solution for real-world cybersecurity challenges.

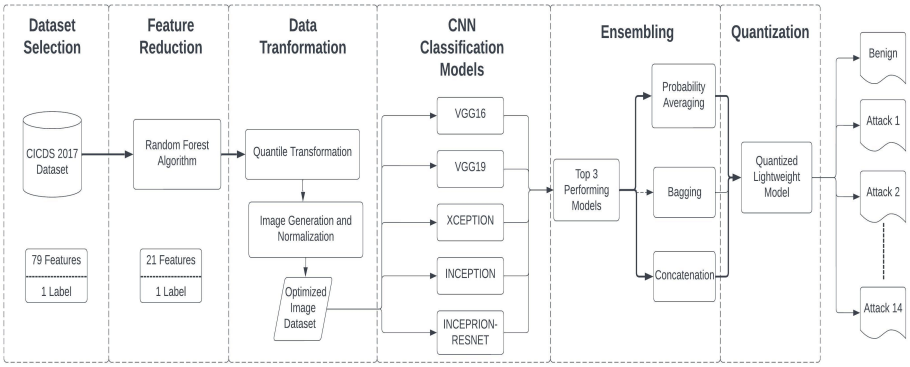

Fig. 1. Proposed Framework

3.2 Dataset Description

Although KDD-99 [9], CAIDA [10], ISCX2012 [11], and Kyoto [12] are useful datasets, they have specific drawbacks that make it difficult to use them to meet modern intrusion detection concerns. For instance, issues such as outdated

data, biases, and limited representation of certain attack types might compromise their suitability for comprehensive intrusion detection model training. The CICIDS2017 dataset [13], on the other hand, has become well-known for its capacity to get beyond these restrictions and offer a more precise and current representation of actual cyberattacks [14].

The CICIDS2017 dataset, a collaborative effort by the University of New Brunswick (UNB) and the Canadian Institute for Cybersecurity (CIC) [15], provides a contemporary and comprehensive resource for intrusion detection research. It captures both normal and various attack scenarios over five days.

3.3 Data Analysis and Preprocessing

The CICIDS2017 dataset is spread out into eight files, each containing distinct subsets or categories of network traffic features, facilitating a comprehensive analysis of diverse aspects of cybersecurity threats. Handling individual files is a laborious process. As a result, we merged those files into a single file that had 28,37,043 occurrences of every file combined.

Firstly, potential infinite values were replaced with NaN to address irregularities. Then, missing values were handled using the SimpleImputer strategy, with a mean-based imputation approach. This step is essential for maintaining dataset integrity and facilitating downstream analysis. The dataset was imputed to handle any missing or inconsistent values. Additionally, we noted that there are 203 instances of missing information and 2,88,602 instances of missing class labels in the dataset which were handled while merging.

The CICIDS2017 dataset is also prone to high-class imbalance [16]. High-class imbalance in a dataset occurs when one or more classes have significantly fewer instances compared to the majority class, posing challenges for machine learning models in accurately predicting minority class instances. To address the high-class imbalance in the dataset, our approach involved targeted undersampling. Our goal was to achieve a more balanced distribution across all classes by tactically reducing the number of instances in the majority class.

3.4 Feature Selection using Random Forest

Random Forest [17] is a powerful algorithm for classification and regression tasks [18]. It creates multiple decision trees using bootstrapped samples and aggregates their predictions. The algorithm randomly selects features at each node, and the final output is based on majority voting. It offers reliable performance with parameters like the number of trees, feature sampling, and minimum node size. Notably, Random Forest inherently performs feature selection during training, aiding in model interpretability. The out-of-bag (OoB) error [19], is computed by utilizing samples that were not employed in the development of each specific tree. This method provides an effective way to estimate the algorithm's classification accuracy without the necessity of a separate test set. The Random Forest algorithm evaluates feature importance through measures like Gini and

permutation importance measures (PIM), enhancing its capabilities for feature selection and model interpretation.

The following represents the mathematical interpretations for Out-of-bag (OOB) error (Equation 1) and permutation importance measure (PIM) (Equation 2) used in the RF classifier:

$$\text{OoB Error} = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{y}_i) \tag{1}$$

$$\text{PIM} = \frac{1}{N}\sum_{i=1}^{N} \left[L(y_i, \hat{y}_{\text{permuted}_i}) - L(y_i, \hat{y}_i)\right] \tag{2}$$

Here, N denotes the total number of observations, y_i represents the true class of the i-th observation, $\hat{y}_{\text{permuted}_i}$ signifies the predicted class of the i-th observation with the permuted feature, $\hat{y}_i$ indicates the predicted class of the i-th observation, and L stands for the loss function typically employed in classification tasks.

Our feature selection procedure effectively whittled down the original set of 79 characteristics to a more manageable 22, highlighting the algorithm's capacity to identify and rank important traits. We achieved this by utilizing Random Forest. Finally, after data preprocessing and feature elimination, the dataset was reduced to 6,55,364 records and 22 features (21 features + 1 label).

3.5 Data Transformation

The dataset undergoes standardization, transforming numerical features to [0, 1] using Quantile Transformation (Equation 3) [20], and scaling by a factor of 255. This prepares the data for CNN models by converting records into 21x21 pixel images with three color channels. Flattening rows into arrays, the data is checked for conformity, and necessary padding or resizing is applied. Resulting arrays are saved as images in 'image_path', linked to DataFrame indices and attack labels. This structured approach ensures data integrity, forming a foundation for supervised learning in ML models. Images are then resized to 224x224 pixels for enhanced learning efficiency.

$$F_X(x) = Q_Y(p) \tag{3}$$

Here, $F_X(x)$ is the empirical cumulative distribution function (ECDF) of the original data, and $Q_Y(p)$ is the quantile function of the desired output distribution. Equation 3 represents the transformation of a data point x in the original dataset to its quantile in the specified output distribution.

After data preprocessing and data transformation, the final set of images is saved as the input for CNN-based IDS. In Fig. 2, the representative samples for each attack are shown.

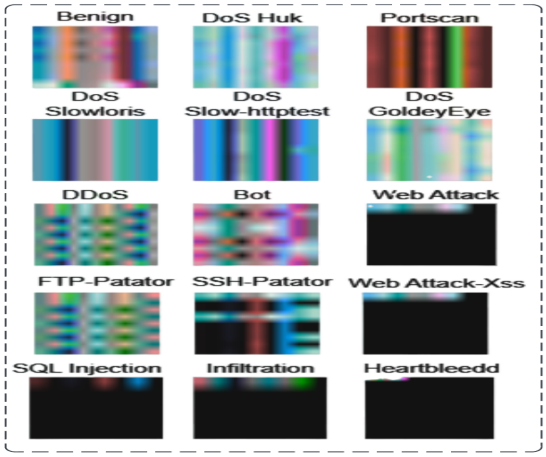

Fig. 2. Sample Images for Every Attack

3.6 Deep Convolutional Neural Networks

One of the most popular deep learning models for image classification and recognition tasks is the Convolutional Neural Network (CNN) [21]. CNNs provide smooth integration by processing pictures immediately, doing away with the need for laborious feature extraction and data reconstruction procedures. The CNN architecture, which consists of convolutional, pooling, and fully connected layers, extracts feature patterns smoothly, minimizes data complexity in the pooling layers, and generates correct outputs through the fully-connected layers. CNNs are a vital tool for tasks involving images because of their architecture, which enables automated feature extraction while maintaining important information and avoiding overfitting.

We utilize the powerful capabilities of VGG16, VGG19 [22], Xception [23], Inception [24], and InceptionResnet [25] as our fundamental CNN models in our proposed architecture. These models were selected based on their proven performance in a variety of picture categorization tasks. Every model has its advantages; VGG16 contains three fully connected layers and five convolutional layer blocks, while VGG19 adds three more convolutional layers. Xception, which uses depthwise separable convolutions to minimize memory needs, complements the Inception network, which is renowned for its ability to reduce dimensionality and extract a variety of features. Incorporating ResNet's residual connections, InceptionResNet excels in image classification, yet mandates heightened computational workloads and necessitates more extensive memory allocations in contrast to the Inception model.

Finally, after training on the five distinguished CNN models on CICIDS2017 Dataset, top-3 performing were selected to construct the ensemble model which is discussed in the next section.

3.7 Proposed Ensemble Learning Technique

Ensembling is a potent machine learning strategy in which the combined use of several models, sometimes with distinct architectures or trained on various subsets of data, improves overall prediction performance.

Our ensemble framework consists of three models, each contributing distinct strategies for improved predictive performance. The concatenation model [26] merges high-level features from three best-performing CNN models, incorporating a dropout layer to prevent redundancy. The probability averaging [27] model refines predictions by averaging probability distributions from individual base models, promoting stability, and mitigating the impact of outliers. The bagging model [28] introduces diversity by training multiple instances of each base model on different data subsets, with ReLU activation in the hidden layers. These models collectively enhance predictive power.

Common to all models are the activation functions—ReLU in the hidden layers to introduce non-linearity— and softmax activation in the output layer for generating probability distributions [29]. The categorical cross-entropy loss function is uniformly applied, optimizing training across the ensemble and ensuring a consistent and effective approach for accurate predictions. This unified use of activation and loss functions strengthens the overall ensemble, providing a robust solution for diverse predictive tasks.

The utilization of the softmax function is common in the final layer of neural networks when dealing with multi-class classification tasks. It transforms the raw output scores (logits) into probabilities. The formula for the softmax function is given in Equation 4 below.

$$\text{Softmax}(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{4}$$

In this context, z_i refers to the raw score corresponding to class i, while K signifies the overall number of classes.

The categorical cross-entropy loss, frequently employed in these scenarios, assesses the disparity between the predicted probability distribution and the actual distribution of the target. The mathematical expression for categorical cross-entropy is depicted in Equation 5.

$$\text{Categorical Crossentropy} = -\sum_{i=1}^{K} y_i \cdot \log(\hat{y}_i) \tag{5}$$

Here, y_i corresponds to the true probability assigned to class i, $\hat{y}_i$ signifies the predicted probability associated with class i, and K represents the total number of classes. The summation is calculated across all classes.

3.8 Quantization for Edge Deployment

In the context of edge deployment, we optimized our ensemble model through quantization, a technique aimed at reducing the model's precision for a more efficient deployment on resource-constrained edge devices [30].

To achieve this, we employed post-training quantization using TensorFlow and Keras [31]. This method involves quantizing the model's weights and activations after the training process, allowing us to achieve a balance between model size reduction and preserved accuracy. The quantized models, individually and when ensembled, exhibited notable efficiency gains, making them well-suited for deployment in edge environments where computational resources are limited.

$$\hat{W}_i = \text{Round}\left(\frac{W_i}{\text{Scale}}\right) \times \text{Scale} \tag{6}$$

Equation 6 describes the quantization process. Here, W_i is the original weight or activation, Round is the rounding function, and Scale is the scaling factor. Scaling ensures that quantized values are adapted to a reduced bit-width while preserving accuracy and aligning with the original value range.

4 Performance Evaluation

4.1 Experimental Setup

The implementation was conducted using Scikit-learn, TensorFlow, and Keras library from the Python 3 environment on the Colab platform. The proposed DL models were trained under the Google Compute Engine with 12.7 GB RAM and T4 GPU with 15 GB space. The testing was done on HP Pavilion Laptop 15 with 8 GB memory.

The presented framework undergoes evaluation using the well-established CICIDS2017 network security dataset, as detailed in section 3.2. To assess the effectiveness of the proposed models, a five-fold cross-validation strategy is employed, serving to mitigate issues related to over-fitting and biased outcomes. Given the inherent imbalance in network traffic data, characterized by a limited proportion of attack samples, performance evaluation relies on four distinct metrics: accuracy, precision, recall, and F1-scores.

4.2 Results and Discussion

The CNN models were fine-tuned using the ADAM optimizer [32], adjusting critical hyperparameters to optimize the learning process. Key parameters like learning rate, momentum, and epsilon were carefully chosen to strike a balance between training efficiency and model accuracy. The careful choice of these hyperparameters significantly influenced the attainment of the targeted convergence and overall performance of the models.

Table 1 displays the results of different models compared with other recent IDS models. The optimized base CNN models achieved an accuracy of 97.8% to 98.4% after implementing data transformation and RF feature selection. The proposed ensembling models also achieve a higher accuracy of up to 98.66%. The three ensembled models also outperform other recent methods in the literature [13].

Table 1. Performance Metrics for Different Models

Models	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1 (%)**	**Classes**
KNN [13]	96.30	96.20	93.70	96.30	-
RF [13]	98.82	98.80	99.95	98.80	-
Concatenation [7]	99.89	99.90	99.89	99.89	6
Confidence Avg. [7]	99.92	99.92	99.92	99.92	6
DCNN	97.80	96.85	97.80	97.23	15
VGG16	98.30	97.53	98.30	97.83	15
VGG19	98.30	98.62	98.30	97.87	15
Inception	98.35	97.54	98.35	97.88	15
InceptionResnet	**98.45**	**98.73**	**98.45**	**98.01**	**15**
Bagging (Ensemble)	98.36	98.00	98.36	98.00	15
Concatenation (Ensemble)	**98.66**	**98.00**	**98.66**	**98.00**	**15**
Prob. Avg. (Ensemble)	**98.58**	**99.00**	**98.58**	**98.00**	**15**

While the Concatenation method proposed by Li Yang *et al.* [7] has an accuracy of up to 99.925% for 6-class classification, it is noteworthy that our ensemble strategy achieves an accuracy of 98.66% at 15-class classification, which supports our reasons for DCNN, RF feature elimination, and data transformation being employed in our framework. Additionally, some of our models also had a lower training time while delivering an accuracy up to 99%.

In examining the post-training quantization outcomes, two notable models, InceptionResnet and Concatenation (Ensembled), reveal intriguing nuances in navigating the delicate trade-off between accuracy preservation and size reduction. InceptionResnet exhibits a subtle trade-off with a minor accuracy dip from 98.45% to 97.38%, yet concurrently achieves a commendable size reduction of 474MB to 53.3MB. Similarly, the Concatenation (Ensembled) model navigates this subtlety adeptly, showcasing a subtle accuracy reduction from 98.66% to 98.13%, accompanied by a significant size reduction of 50.21%.

Table 2. Model Accuracy Comparison with and without Quantization

Model	**Accuracy w/o Quantization (%)**	**Accuracy with Quantization (%)**	**Size Reduction (%)**
VGG19	98.3	97.76	83.06
Inception	98.35	97.7	83.19
InceptionResnet	**98.45**	**97.38**	**88.74**
Concatenation (Ensembled)	**98.66**	**98.13**	**50.21**

This nuanced perspective, as reflected in the provided Table 2, underscores the intricacy of navigating the trade-off between model performance and efficiency during quantization. Striking this balance becomes pivotal, especially in resource-constrained settings such as edge deployments.

5 Conclusion

Finally, this study effectively creates an efficient Intrusion Detection System optimized for edge computing. The ensemble learning approach, using Deep Neural Networks combined with Random Forest for feature reduction, not only enhances feature selection but also significantly improves the IDS's performance. Post quantization, our proposed ensembled model exhibits an accuracy of 98.13%, a marginal decrease from its original 98.66%, along with a size reduction of over 50%. The InceptionResnet model maintains a high accuracy of 97.38% after quantization, down from 98.45%, with a substantial size reduction of approximately 88.74%. These results demonstrate the system's capability to balance computational efficiency with exceptional detection accuracy, even in resource-limited environments. Such innovations underscore the potential for advancing cybersecurity, pointing towards a promising future for lightweight, yet powerful, cybersecurity technologies.

Looking ahead, the future scope of research includes further exploration of advanced quantization techniques, aiming to optimize model size reduction without compromising detection accuracy. Additionally, investigating the adaptability of the proposed IDS for diverse edge computing environments and evaluating its performance across various real-world scenarios could contribute to refining and expanding its applicability. These avenues of research promise to propel the field of lightweight yet powerful cybersecurity technologies, offering innovative solutions to the evolving challenges of intrusion detection in resource-constrained settings.

References

1. P. Parkar and A. Bilimoria, "A Survey on Cyber Security IDS using ML Methods," 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2021, pp. 352-360, doi: 10.1109/ICICCS51141.2021.9432210.
2. U. Bashir and M. Chachoo, "Intrusion detection and prevention system: Challenges & opportunities," 2014 International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2014, pp. 806-809, doi: 10.1109/IndiaCom.2014.6828073.
3. Hoang, Trong-Minh & Thi, Trang-Linh & Quy, Nguyen. (2023). A Novel Distributed Machine Learning Model to Detect Attacks on Edge Computing Network. Journal of Advances in Information Technology. 14. 10.12720/jait.14.1.153-159.
4. Hajj, Suzan, et al. "Anomaly-based intrusion detection systems: The requirements, methods, measurements, and datasets." Transactions on Emerging Telecommunications Technologies 32.4 (2021)
5. Ahmad, R., Alsmadi, I., Alhamdani, W. et al. Zero-day attack detection: a systematic literature review. Artif Intell Rev 56, 10733–10811 (2023). https://doi.org/10.1007/s10462-023-10437-z
6. L. Yang and A. Shami, "A Transfer Learning and Optimized CNN Based Intrusion Detection System for Internet of Vehicles," ICC 2022 - IEEE International Conference on Communications, Seoul, Korea, Republic of, 2022, pp. 2774-2779, doi: 10.1109/ICC45855.2022.9838780.

7. P. Spadaccino and F. Cuomo, “Intrusion Detection Systems for IoT: opportunities and challenges offered by Edge Computing and Machine Learning,” arXiv (Cornell University), Dec. 2020, doi: 10.48550/arxiv.2012.01174.
8. Hasan, Md Al Mehedi, et al. ”Feature selection for intrusion detection using random forest.” Journal of Information Security 7.3 (2016): 129-140.DOI: 10.4236/jis.2016.73009
9. M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani, ”A detailed analysis of the KDD CUP 99 data set,” 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, 2009, pp. 1-6, doi: 10.1109/CISDA.2009.5356528.
10. Vaishali Shirsath, May 12, 2023, ”CAIDA UCSD DDoS 2007 Attack Dataset”, IEEE Dataport, doi: https://dx.doi.org/10.21227/dvp9-s124.
11. S. Soheily-Khah, P. -F. Marteau and N. Béchet, ”Intrusion Detection in Network Systems Through Hybrid Supervised and Unsupervised Machine Learning Process: A Case Study on the ISCX Dataset,” 2018 1st International Conference on Data Intelligence and Security (ICDIS), South Padre Island, TX, USA, 2018, pp. 219-226, doi: 10.1109/ICDIS.2018.00043.
12. Song, Jungsuk et al. “Statistical analysis of honeypot data and building of Kyoto 2006+ dataset for NIDS evaluation.” International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (2011).https://doi.org/10.1145/1978672.1978676
13. Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorba-ni, “Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization”, 4th International Conference on Information Systems Security and Privacy (ICISSP), Portugal, January 2018
14. R. Panigrahi and S. Borah, “A detailed analysis of cicids2017 dataset for designing intrusion detection systems,” vol. 7, pp. 479–482, 01 2018.
15. (2018) Intrusion Detection Evaluation Dataset. https://www.unb.ca/cic/datasets/ids-2017.html
16. Galar, A. Fernandez, E. Barrenechea, H. Bustince and F. Herrera, ”A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approach-es,” in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 42, no. 4, pp. 463-484, July 2012. doi: 10.1109/TSMCC.2011.2161285
17. Breiman, L. Random Forests. Machine Learning 45, 5–32 (2001). https://doi.org/10.1023/A:1010933404324
18. Liaw, A., & Wiener, M. C. (2007). ”Classification and Regression by randomForest.” https://api.semanticscholar.org/CorpusID:3093707.
19. Janitza S, Hornung R. On the overestimation of random forest’s out-of-bag error. PLoS One. 2018 Aug 6;13(8):e0201904. doi: 10.1371/journal.pone.0201904. PMID: 30080866; PMCID: PMC6078316.
20. Abdulraheem, Mohammad Hamid, and Najla Badie Ibraheem. ”A detailed analysis of new intrusion detection dataset.” Journal of Theoretical and Applied Information Technology 97.17 (2019): 4519-4537 https://www.jatit.org/volumes/Vol97No17/4Vol97No17.pdf
21. Mohammadpour, L.; Ling, T.C.; Liew, C.S.; Aryanfar, A. A Survey of CNN-Based Network Intrusion Detection. Appl. Sci. 2022, 12, 8162. https://doi.org/10.3390/app12168162
22. Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. 3rd International Conference on Learning Representations (ICLR 2015).

23. Chollet, François. “Xception: Deep Learning with Depthwise Separable Convolutions.” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016). https://doi.org/10.48550/arXiv.1610.02357
24. Szegedy, Christian et al. “Going deeper with convolutions.” 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2014). https://doi.org/10.48550/arXiv.1409.4842
25. Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. 2017. Inception-v4, inception-ResNet and the impact of residual connections on learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI’17). AAAI Press. https://doi.org/10.48550/arXiv.1602.07261
26. M. Rahimzadeh and A. Attar, “A modified deep convolutional neural network for detecting COVID-19 and pneumonia from chest X-ray images based on the concatenation of Xception and ResNet50V2”, Informatics Med., 2020
27. Treisman, M. (1998). Combining information: Probability summation and probability averaging in detection and discrimination. Psychological Methods, 3(2), 252.
28. Breiman, Leo. ”Bagging predictors.” Machine learning 24 (1996): 123-140.
29. B. Ding, H. Qian and J. Zhou, ”Activation functions and their characteristics in deep neural networks,” 2018 Chinese Control And Decision Conference (CCDC), Shenyang, China, 2018, doi: 10.1109/CCDC.2018.8407425.
30. Chenna, D. EDGE AI: QUANTIZATION AS THE KEY TO ON-DEVICE SMARTNESS. Journal ID, 4867, 9994.
31. LAVAGNO, L., LAZARESCU, M. T., & BATTIATO, L. Tensorflow-driven neural network quantization for resource-constrained devices. https://webthesis.biblio.polito.it/16621/1/tesi.pdf
32. Kingma, Diederik P. and Jimmy Ba. “Adam: A Method for Stochastic Optimization.” CoRR abs/1412.6980 (2014): n. pag. https://doi.org/10.48550/arXiv.1412.6980