Automatic Attack Detection in Wi-Fi Networks

Fărcășanu Tudor Andrei
Faculty of Mathematics and Computer Science
University of Bucharest
andrei.farcasanu2002@gmail.com
tudor-andrei.farcasanu@s.unibuc.ro

Abstract

This paper presents methods for detecting and classifying Wi-Fi attacks using deep learning techniques, employing AWID3, a dataset designed for developing Wi-Fi intrusion detection systems. The dataset was processed to reduce dimensionality and remove irrelevant information for layer 2 attack detection. Considering works in the same field that used earlier versions of AWID, both classical and autoencoder neural networks were developed and trained for binary traffic classification and multi-class attack classification scenarios. Finally, the models and strategies were evaluated based on the average F1 score obtained in the multi-class attack classification scenario, experimentally demonstrating the superior performance of the approach based on classical neural networks.

1 Introduction

Wi-Fi networks have become a cornerstone of modern communication infrastructure, facilitating ubiquitous access to the internet and enabling a plethora of wireless devices to interconnect seamlessly. However, the widespread adoption of Wi-Fi technology has also made it a prime target for a variety of cyber-attacks. These attacks can compromise network integrity, confidentiality, and availability, posing significant risks to both individual users and organizations. Consequently, intrusion detection systems (IDS) that can effectively identify and mitigate these threats are needed.

Artificial Intelligence has made substantial contributions to the development of advanced IDS. Techniques such as deep learning have shown great promise in enhancing the accuracy and efficiency of attack detection and identification mechanisms, with the ability to learn complex patterns and representations from large datasets.

This paper addresses the critical challenge of detecting Wi-Fi attacks by leveraging deep learning techniques. This study aims to develop and evaluate methods that can accurately classify various types of Wi-Fi attacks by using the *Aegean Wi-Fi Intrusion Dataset* (AWID3)[2], which is specifically designed for use in the development of intrusion detection systems.

In the broader context of the Artificial Intelligence field, this work contributes to the ongoing efforts to harness deep learning for cybersecurity applications. The methodologies and findings presented in this paper contribute to the state of the art in Wi-Fi intrusion detection, by demonstrating the efficacy of deep learning models in a practical setting.

Main contributions:

- Development and evaluation of "classical" neural network and autoencoder-based approaches for Wi-Fi attack detection and classification.
- Preprocessing and feature engineering of the AWID3 [2] dataset to focus on relevant layer 2 OSI attack information.

2 Related Work

Previous research in Wi-Fi attack detection has utilized earlier versions of the AWID dataset. The specific literature gap this paper aims to address is the lack of studies using the newest version of the AWID dataset, AWID3, for Wi-Fi intrusion detection.

Some of the works that used older versions of AWID include Wang et al. (2019) [3] who used two types of deep learning models to perform attack classification, namely deep neural networks (DNN) and stacked autoencoders (SAE). Their study achieved reasonably high accuracy in classifying the types of attacks present in AWID2.

Basnet and Kholidy (2020) [1] attempted Wi-Fi attack classification using various machine learning techniques on AWID2. They utilized both traditional machine learning algorithms, including Random Forest and Decision Trees, as well as deep learning models such as "classical" deep neural networks and Long Short-Term Memory (LSTM) networks, achieving high accuracy in both binary and multi-class attack classification.

3 Approach

3.1 Classical Neural Networks

I implemented and trained neural network architectures for both binary traffic classification (normal vs. malicious) and multi-class attack classification (8 categories). Initially, I adapted an architecture inspired by [1], which contained Dropout layers, for a total of 5:

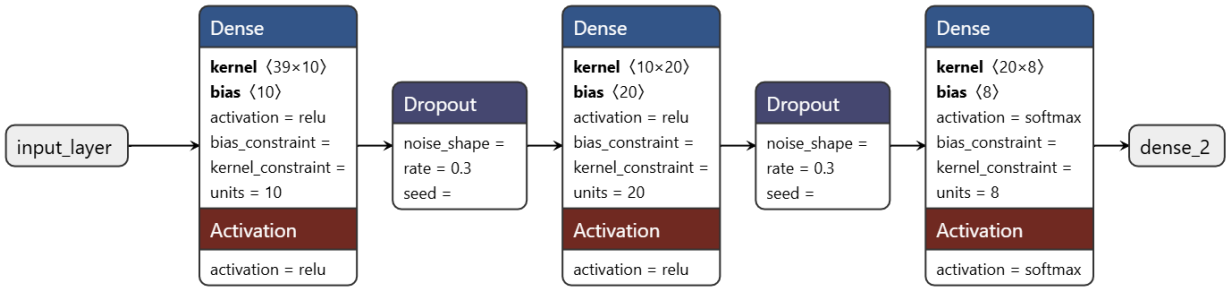

Figure 1: Initial architecture - Output size 8, equal to the number of categories, for multi-class attack classification, softmax activation function

Through experimentation, I found that reducing the weight of Dropout layers improved results, and subsequently eliminated them. The final architecture consists of three layers, with the input layer matching the number of features (39) and the output layer adapted for each classification task:

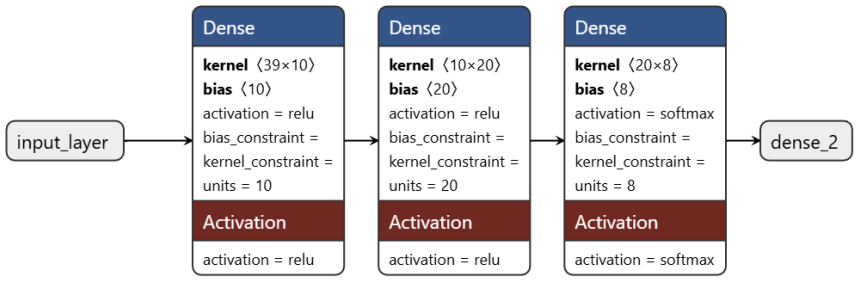

Figure 2: Final architecture - Multi-class classification, without Dropout

3.2 Autoencoders

I trained multiple autoencoder networks, one for each traffic category, using only the data labeled as that specific category. I initially experimented with a 15-layer architecture that compressed the data from the initial dimension to 32, 16, 8, and finally 4 units in the coding layer, with Dropout layers interspersed. This architecture showed poor performance. I then tested an intermediate architecture with 7 layers, removing the Dropout layers, which improved its reconstruction errors.

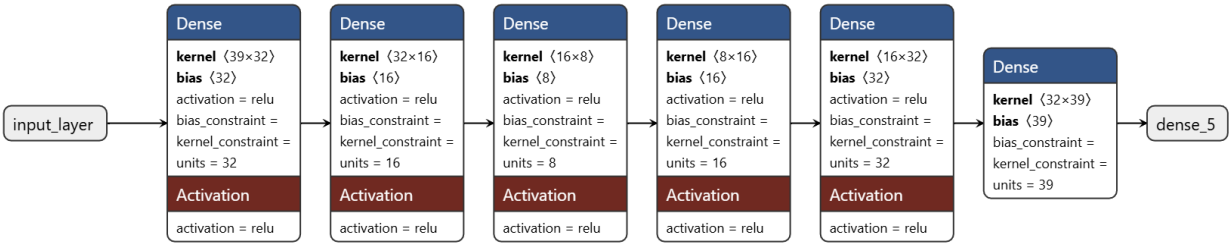

Figure 3: Intermediate architecture - Without Dropout layers

The final autoencoder architecture consists of 5 layers, with the encoder compressing the data from the initial dimension to 32 units and then to 16 units in the code layer:

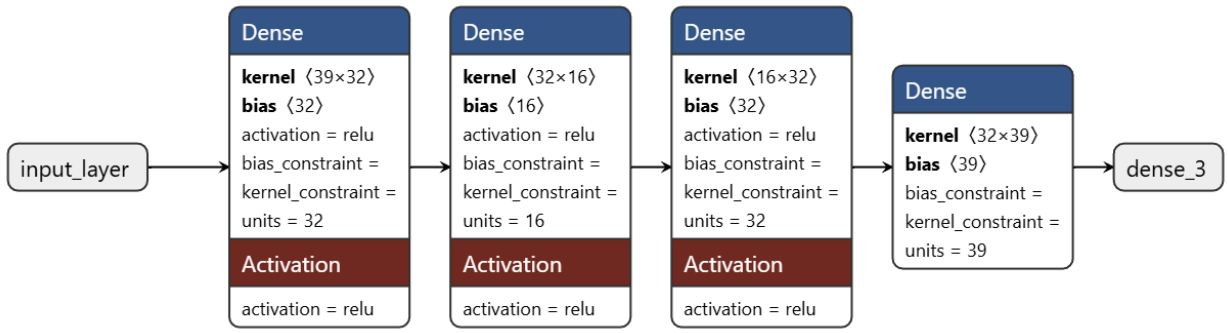

Figure 4: Final autoencoder architecture

The reduction in the number of layers and the removal of Dropout layers led to improved reconstruction errors for the autoencoders, as shown in the following table:

Table 1: Comparison of reconstruction errors

Class	Best Epoch	Intermediate model	Best Epoch	Final model
Normal	1746	1.18550e-02	1560	2.08367e-03
Kr00k	5577	7.20589e-05	4664	3.52272e-05
Evil_Twin	6377	5.00988e-03	4352	9.29793e-04
Disas	9994	4.08687e-04	9989	4.09299e-05
Krack	10000	2.41950e-02	9999	8.35681e-03
Deauth	10000	6.51598e-04	10000	4.82967e-05
(Re)Assoc	10000	1.05994e-03	10000	6.16459e-04
RogueAP	9500	4.72845e-06	7556	8.07956e-06

For the autoencoder approach, I developed two classification methods:

1. **Threshold-based classification**: I searched for optimal thresholds for the reconstruction errors of each autoencoder on the validation set, maximizing the F1 score.

2. **Normal distribution-based classification**: I calculated the parameters of normal distributions for the reconstruction errors of each class on the validation set and used these to compute probabilities for test instances.

4 Experimental Analysis

4.1 Dataset and Preprocessing

I utilized AWID3 (Aegean Wi-Fi Intrusion Dataset) [2], which is designed for developing Wi-Fi intrusion detection systems. This dataset contains attacks on the WPA2 security protocol in networks with PMF (Protected Management Frames) enabled and in environments using EAP (Extensible Authentication Protocol).

The initial dataset contained 254 features. I performed the following preprocessing steps:

- Removed 181 columns containing data corresponding to layers above level 2 on the OSI stack or completely null across all files.
- Deleted samples with null labels or mostly null properties.
- Filled missing values in remaining columns with null elements or contextually appropriate values.
- Converted text-based columns to boolean or numeric values where appropriate.
- Encoded MAC addresses using the Python macaddress module.
- Transformed cryptographic elements specific to WPA2 into numeric format using a hashing technique.

After preprocessing, our final dataset contained 15,574,909 records, of which 467,505 (approximately 3%) are anomalies. The distribution of attack types is shown in Table 2.

Category	**Total Count**	**Proportion**
Normal	15107404	96.99%
Kr00k	191803	1.23%
Evil_Twin	104827	0.67%
Disas	75131	0.48%
Krack	49990	0.32%
Deauth	38942	0.25%
(Re)Assoc	5502	0.035%
RogueAP	1310	0.0084%

Table 2: Distribution of attack types in the used data

The dataset was split into training, validation, and test sets using stratified sampling to maintain the class distribution across subsets. I then standardized the data using StandardScaler from scikit-learn.

4.2 Classical Neural Network Results

I evaluated the classical neural network models for both binary and multi-class classification tasks. Tables 3 and 4 show the classification reports for multi-class and binary classification, respectively, comparing models with and without dropout layers.

Table 3: Multi-class attack classification report

Class	Support	Model without Dropout layers			Model with Dropout rate 0.3		
		Precision	Recall	F1-score	Precision	Recall	F1-score
Normal	1510740	1.00	1.00	1.00	1.00	1.00	1.00
Kr00k	19181	1.00	1.00	1.00	1.00	1.00	1.00
Evil_Twin	10483	1.00	1.00	1.00	0.99	0.96	0.98
Disas	7513	1.00	1.00	1.00	1.00	0.99	0.99
Krack	4999	0.96	0.93	0.94	0.86	0.79	0.82
Deauth	3894	1.00	1.00	1.00	0.78	1.00	0.88
(Re)Assoc	550	1.00	1.00	1.00	0.79	0.63	0.70
RogueAP	131	1.00	1.00	1.00	0.00	0.00	0.00
average		0.99	0.99	0.99	0.80	0.80	0.80
weighted avg		1.00	1.00	1.00	1.00	1.00	1.00

Table 4: Binary traffic classification report

Class	Support	Model without Dropout layers			Model with Dropout rate 0.3		
		Precision	Recall	F1-score	Precision	Recall	F1-score
Normal	1510741	1.00	1.00	1.00	1.00	1.00	1.00
Malicious	46750	0.99	1.00	0.99	0.98	0.99	0.98
average		1.00	1.00	1.00	0.99	0.99	0.99
weighted avg		1.00	1.00	1.00	1.00	1.00	1.00

4.3 Autoencoder Results

For the autoencoder approach, I implemented two classification methods: threshold-based and normal distribution-based. Table 5 provides a comparison of the classification reports for both autoencoder-based methods.

Table 5: Attack classification report for autoencoder methods

Class	Support	Threshold classification			Normal dist. classification		
		Precision	Recall	F1-score	Precision	Recall	F1-score
Normal	1510741	1.00	1.00	1.00	1.00	1.00	1.00
Kr00k	19180	1.00	1.00	1.00	1.00	1.00	1.00
Evil_Twin	10483	0.99	1.00	0.99	0.99	1.00	0.99
Disas	7513	0.94	1.00	0.97	0.98	1.00	0.99
Krack	4999	0.84	0.83	0.84	0.39	0.05	0.10
Deauth	3894	0.99	1.00	0.99	1.00	1.00	1.00
(Re)Assoc	550	0.75	1.00	0.86	0.92	0.99	0.96
RogueAP	131	1.00	1.00	1.00	1.00	0.98	0.99
average		0.94	0.98	0.96	0.91	0.88	0.88
weighted avg		1.00	1.00	1.00	0.99	1.00	1.00

4.4 Comparison of Methods

Table 6 provides a comparison of F1-scores across all implemented methods.

Table 6: Comparison of F1-scores

Class	Classical Neural Network F1-score	AE - thresholds F1-score	AE - normal dist. F1-score
Normal	1.00	1.00	1.00
Kr00k	1.00	1.00	1.00
Evil_Twin	1.00	0.99	0.99
Disas	1.00	0.97	0.99
Krack	0.94	0.84	0.10
Deauth	1.00	0.99	1.00
(Re)Assoc	1.00	0.86	0.96
RogueAP	1.00	1.00	0.99
average	0.99	0.96	0.88

These results demonstrate that the classical neural network approach outperforms the autoencoder methods in terms of overall classification accuracy and F1-scores for individual attack categories. The threshold-based classification method for autoencoders performed better than the normal distribution-based method, particularly for the Krack attack class.

5 Discussion

These results demonstrate that the classical neural network approach outperforms the autoencoder methods in terms of overall classification accuracy and F1-scores for individual attack categories. The neural network achieved near-perfect classification for most attack categories, with only a slight decrease in performance for the Krack attack (F1 score of 0.94).

The autoencoder approach, while effective, showed lower performance compared to the classical neural network. This difference may be attributed to the autoencoder's unsupervised nature, which might not capture class-specific features as effectively as the supervised neural network.

The threshold-based classification method for autoencoders performed better than the normal distribution-based method, particularly for the Krack attack class. This indicates that reconstruction error thresholds might capture attack signatures more effectively than assuming normal distributions of errors.

Limitations of this study include the imbalanced nature of the dataset (normal traffic significantly outnumbers attack instances) and the availability of a single Wi-Fi focused dataset (AWID3). While AWID3 is comprehensive, testing the models on additional datasets would further validate their ability to generalize.

6 Conclusion

This paper demonstrates the effectiveness of deep learning techniques for Wi-Fi attack detection and classification, with classical neural networks outperforming autoencoder-based approaches. Our study focused on the AWID3 dataset, which we preprocessed to enhance its suitability for layer 2 attack detection.

The main contributions of this work include the development and comparison of classical neural networks and autoencoder-based methods for Wi-Fi attack classification, achieving near-perfect classification, as well as effective preprocessing of the AWID3 dataset for the same purpose.

Future directions for this research include:

1. Applying other anomaly detection methods, both supervised and unsupervised, to the AWID3 dataset.
2. Developing a real-time attack detection and classification system that can run on IoT devices with limited processing capacity, such as routers or smart home hubs.

This work contributes to enhancing Wi-Fi network security in an era of increasing connectivity and cyber threats, providing a foundation for more robust and efficient intrusion detection systems.

References

[1] Diwash Bikram Basnet and Hisham A.; Advisor Kholidy. An empirical wi-fi intrusion detection system, 2020. URL `http://hdl.handle.net/20.500.12648/1603`.

[2] E. Chatzoglou, G. Kambourakis, and C. Kolias. Empirical evaluation of attacks against ieee 802.11 enterprise networks: The awid3 dataset. *IEEE Access*, 9:34188–34205, 2021. doi: 10.1109/ACCESS.2021.3061609.

[3] Shaoqian Wang, Bo Li, Mao Yang, and Zhongjiang Yan. Intrusion detection for wifi network: A deep learning approach. In Jiann-Liang Chen, Ai-Chun Pang, Der-Jiunn Deng, and Chun-Cheng Lin, editors, *Wireless Internet*, pages 95–104, Cham, 2019. Springer International Publishing. ISBN 978-3-030-06158-6.