

Detection and Anticipation of Cyberattacks in Institutional Networks Using Machine Learning Techniques

Wildote Valério Valente¹

¹Academia Militar Marechal Samora Machel

Publication Date: 2026/10/08

Abstract: Mozambican institutional networks face a growing volume of cyberattacks, with more than 173,000 incidents recorded in 2024, while national incident response capacity is still being consolidated. This paper experimentally evaluates machine learning techniques for intrusion detection and for its anticipation, defined as classifying a connection as malicious using only the information available at its start. The public UNSW-NB15 dataset (175,341 training and 82,332 test records, nine attack categories) was used to compare four algorithms: Logistic Regression, Random Forest, XGBoost and a Multilayer Perceptron. Two scenarios were defined: detection using the 42 full-flow features, and anticipation using a subset of 20 features available at connection set-up. In the detection scenario, XGBoost achieved the highest area under the ROC curve (0.982), with an F1-score of 0.892 and a recall of 98.0%. In the anticipation scenario it retained an AUC of 0.978 and an F1-score of 0.873, showing that most of the discriminative power is already present before the flow ends. The false positive rate at the default threshold (26.5%) was the main operational obstacle, but threshold tuning reduces it to 5% while preserving 91.1% detection. Boosted decision-tree models are therefore suitable components of an intrusion detection system for institutional networks, provided they are locally calibrated and validated on real traffic.

Keywords: Cybersecurity, Intrusion Detection, Machine Learning, XGBoost, UNSW-NB15, Mozambique.

How to Cite: Wildote Valério Valente (2026) Detection and Anticipation of Cyberattacks in Institutional Networks Using Machine Learning Techniques. *International Journal of Innovative Science and Research Technology*, 11(9), 3235-3243. <https://doi.org/10.38124/ijisrt/26sep1439>

I. INTRODUCTION

The digitalization of public services, educational institutions and the defense and security forces has widened the attack surface of institutional networks in Mozambique. According to the Ministry of Communications and Digital Transformation, the country recorded 173,770 cyberattacks in 2024, which led to 1,061 criminal cases, and 36,330 attacks in the first half of 2025 alone [1]. In February 2022, more than 30 government websites were attacked simultaneously, in what was described as the first incident of this magnitude in the country [2]. In the National Cyber Security Index (NCSI), Mozambique ranks 108th, scoring zero on the indicators for critical information infrastructure protection and for research and development [3].

Traditional defenses, such as *firewalls* and signature-based intrusion detection systems (IDS), only identify attacks whose pattern is already known. Faced with new or obfuscated variants, and with the traffic volume of a modern network, these approaches become insufficient. Machine learning techniques offer an alternative: instead of manually written rules, the system learns from labeled examples to distinguish

normal from malicious traffic, and can generalize to attacks that do not exactly match the training examples [4], [5].

For an institution with limited specialized staff, detecting an attack after it has succeeded is of little value. What matters most is recognizing hostile behavior as early as possible, ideally while the connection is being established, so that it can be blocked preventively. This concern connects to the literature on attack projection and prediction [6] and to the intrusion *kill chain* model, according to which disrupting the adversary in the early phases is more effective than reacting in the final ones [7].

In this context, the research problem is stated as follows: to what extent do machine learning techniques make it possible to detect cyberattacks in institutional networks and to anticipate them from the information available at the start of connections? The general objective is to experimentally evaluate the performance of different algorithms on this twofold task. The specific objectives are to (i) compare four families of algorithms in the binary detection of malicious traffic; (ii) quantify the performance loss when the model only has access to features available at connection start; (iii) analyze performance per attack category and identify the most

influential features; and (iv) discuss the implications for deploying an IDS in a Mozambican institutional network.

The paper is organized in six sections. Section II presents the theoretical background and related work. Section III describes the methodology, the dataset and the experimental design. Section IV presents the results, which are discussed in Section V. Section VI presents the conclusions, limitations and directions for future work.

II. BACKGROUND AND RELATED WORK

➤ *Intrusion Detection Systems*

An intrusion detection system is a mechanism that monitors network traffic or system activity in order to identify actions that compromise the confidentiality, integrity or availability of information. The model proposed by Denning [8] established the central idea that intrusions manifest themselves as deviations from usual behavior. Two approaches are traditionally distinguished. Signature-based detection compares traffic with patterns of known attacks and produces few false alarms, but cannot recognize new attacks. Anomaly-based detection builds a model of normal behavior and flags deviations, which allows it to recognize previously unseen attacks at the cost of a higher false positive rate. IDS based on supervised learning occupy an intermediate position: they learn from examples of both classes and generalize to variants close to those observed.

➤ *Machine Learning Applied to Cybersecurity*

The surveys by Buczak and Guven [4] and Ahmad et al. [5] show that virtually every family of algorithms has been tested for intrusion detection, from linear classifiers and decision trees to deep neural networks. Tree-based *ensemble* methods, such as *Random Forest* [9] and *gradient boosting*, particularly in its XGBoost implementation [10], consistently stand out on tabular data such as network flow records, because they handle heterogeneous features, different scales and non-linear interactions well. Neural networks achieve competitive results but require more data, longer training and greater tuning effort. The literature also warns about the pitfalls of this field. Sommer and Paxson [11] argue that intrusion detection differs from other machine learning domains because of the high cost of errors, the variability of real traffic and the difficulty of obtaining representative data. Arp et al. [12] identify recurring problems in security research using machine learning, including data leakage between training and test sets, spurious features that correlate with the class as an artifact of data collection, and evaluation with metrics unsuited to imbalanced data. These warnings guided the experimental design of this work.

➤ *Datasets for Evaluation*

IDS evaluation relies on labeled datasets. KDD Cup 99 and its revised version NSL-KDD were the reference for years, but they contain redundant records and traffic that no longer represents current networks [13]. CICIDS2017 [14] is more recent, but errors have been documented in its labeling and flow extraction [15]. UNSW-NB15, produced at the *Cyber Range Lab* of the University of New South Wales, combines real normal traffic with nine families of modern attacks and

provides official training and test partitions [16], [17]. Ring et al. [18] identify it as one of the most widely used and complete datasets for network-based IDS. Using the same official partition and the 42 features, Kasongo and Sun [19] obtained binary classification test accuracies between 79.6% (Logistic Regression) and 88.1% (decision tree), and 75.6% in multiclass classification with a neural network, which serve as a reference for the results of this study.

➤ *From Detection to Anticipation*

Husák et al. [6] distinguish three forms of anticipatory reasoning in cybersecurity: projection of the next step of an ongoing attack, prediction of the type and target of future attacks, and forecasting of aggregate trends. The approach adopted in this paper is closer to early detection: it seeks to classify a connection as malicious before it ends, using only connection set-up parameters and the context of previous connections. This formulation has direct operational value, because the alert arrives in time to block the connection, and it fits the *kill chain*: the reconnaissance and delivery phases precede exploitation and are the natural point of disruption [7].

➤ *The Mozambican Context*

In 2021 Mozambique approved its National Cyber Security Policy and Strategy, whose implementation is the responsibility of the National Institute of Information and Communication Technologies (INTIC). The country has incident response teams for the Government (CSIRT GOV, run by INAGE) and for the academic network (MoRENET) [2]. A new National Cyber Security Strategy for 2026–2030 is being prepared, which foresees the consolidation of the national CSIRT and mandatory incident reporting [1]. In institutions, including academic and military ones, detection still relies mainly on commercial signature-based tools, with little in-house analytical capacity. An IDS based on locally trained and calibrated models can strengthen that capacity at low cost, since the tools used in this study are open source.

III. METHODOLOGY

➤ *Type of Study*

This is a quantitative, experimental and applied study in which supervised classification algorithms are compared under controlled and reproducible conditions. The experiments were implemented in Python 3.11 with the scikit-learn 1.8 [20] and XGBoost 3.2 libraries, on a virtual machine with 2 processor cores and 8 GB of memory. All algorithms used random seed 42 so that the experiments can be repeated.

➤ *Dataset*

The official UNSW-NB15 partitions were used. Each record corresponds to a network flow described by 42 features, three of them categorical (protocol, service and connection state) and the rest numerical. The features include durations, packet and byte counts, TTL values, TCP windows, *handshake* timings and counters summarizing the previous 100 connections with the same source, destination or service. Table 1 shows the class distribution. The test set is independent of the training set and has a different proportion of attacks (55.1% versus 68.1%), which makes the evaluation more demanding than a simple random split.

Table 1 Distribution of Records Per Category in UNSW-NB15

Category	Description	Train	Test
Normal	Legitimate traffic	56,000	37,000
Generic	Attacks on block ciphers	40,000	18,871
Exploits	Vulnerability exploitation	33,393	11,132
Fuzzers	Random data injection	18,184	6,062
DoS	Denial of service	12,264	4,089
Reconnaissance	Scanning and probing	10,491	3,496
Analysis	Port scans, spam, HTML	2,000	677
Backdoor	Hidden access	1,746	583
Shellcode	Injected code	1,133	378
Worms	Self-replicating code	130	44
Total		175,341	82,332

➤ Preprocessing

Categorical features were converted to integers by ordinal encoding fitted on the training set only (values unseen in training receive a dedicated code). Numerical features with maximum values above 100, which have long-tailed distributions, were transformed with $\log(1 + x)$. All features were then standardized to zero mean and unit variance, with parameters estimated exclusively on the training set to avoid information leakage into the test set. Record identifiers and the attack category were excluded from the explanatory variables.

➤ Experimental Scenarios

Two scenarios were defined. In the detection scenario, the model receives all 42 features, which corresponds to classifying the flow after it has ended. In the anticipation scenario, the model receives only the 20 features known at connection start: protocol and service; source and destination TTL and TCP windows; initial sequence numbers; *handshake* timings (tcprtt, synack and ackdat); the eight context counters computed over previous connections; and the indicator of equal source and destination addresses and ports. All features that are only known at the end of the flow, such as duration, byte and packet volumes, loss rates and final connection state, are excluded. The performance difference between the two scenarios measures the cost of deciding earlier.

➤ Algorithms

Four algorithms representing distinct families were compared. Logistic Regression serves as the linear baseline. Random Forest [9], with 200 trees, combines decision trees trained on random samples. XGBoost [10], with 300 trees of maximum depth 8 and a learning rate of 0.1, builds trees

sequentially, each correcting the errors of the previous ones. The Multilayer Perceptron (MLP), with two hidden layers of 128 and 64 neurons and early stopping, represents neural networks. No exhaustive hyperparameter tuning was performed, since the goal is to compare algorithm families under equivalent conditions rather than to maximize a single result.

➤ Evaluation Metrics

Taking the attack as the positive class, the metrics used were accuracy, precision, recall (detection rate), F1-score, area under the ROC curve (AUC) and false positive rate (FPR), i.e., the proportion of legitimate traffic classified as an attack. The FPR is of particular operational importance, because each false alarm consumes analyst time and, in excess, causes alerts to be ignored. Training time and mean inference time per flow were also measured. In addition, a multiclass XGBoost classifier was trained to identify the category of each attack.

IV. RESULTS

➤ Binary Detection

Table 2 shows the performance of the four algorithms in both scenarios. In the detection scenario, the two tree-based methods obtained the best results. Random Forest achieved the highest accuracy (87.2%) and F1-score (0.894), while XGBoost obtained the highest AUC (0.982) and the lowest FPR (26.5%), with minimal differences between the two. The MLP was slightly behind, and Logistic Regression performed worst, especially in terms of FPR (38.3%), confirming that the boundary between normal and malicious traffic is not linear.

Table 2 Performance of the Algorithms on the Test Set (Decision Threshold 0.5)

Algorithm	Acc.	Prec.	Recall	F1	AUC	FPR
Detection (42 features)						
Logistic Reg.	80.8%	0.755	0.965	0.847	0.937	38.3%
Random Forest	87.2%	0.818	0.987	0.894	0.979	26.9%
XGBoost	87.0%	0.819	0.980	0.892	0.982	26.5%
MLP	85.6%	0.805	0.975	0.882	0.975	28.9%
Anticipation (20 features)						
Logistic Reg.	77.5%	0.724	0.955	0.824	0.886	44.6%
Random Forest	84.5%	0.786	0.988	0.876	0.955	32.9%
XGBoost	84.2%	0.783	0.985	0.873	0.978	33.5%
MLP	82.8%	0.774	0.973	0.862	0.966	34.9%

These values are in line with the literature using the same official partition: Kasongo and Sun [19] report 79.6% for Logistic Regression and 86.7% for a neural network, very close to the values obtained here (80.8% and 85.6%). All algorithms show the same pattern: very high recall (above 95%) and, in the detection scenario, an FPR between 26% and 38%. In other words, almost all attacks are detected, but a significant share of legitimate traffic is flagged as suspicious. This behavior results partly from the distribution shift between training and test sets (the proportion of normal traffic is higher in the test set) and from the default decision threshold of 0.5, an issue analyzed in Section IV-C. The XGBoost confusion matrix shows 44,421

correctly detected attacks and only 911 missed ones, against 9,811 false alarms out of 37,000 legitimate flows.

Fig. 1 compares the F1-score and the FPR in both scenarios. In terms of computational cost, XGBoost was the most efficient of the non-linear models: it trained in 3.8 seconds, against 25.0 for Random Forest and 50.8 for the MLP, and classified each flow in about 2.2 microseconds, which corresponds to hundreds of thousands of flows per second on a single core. These values make real-time use feasible on a modest server.

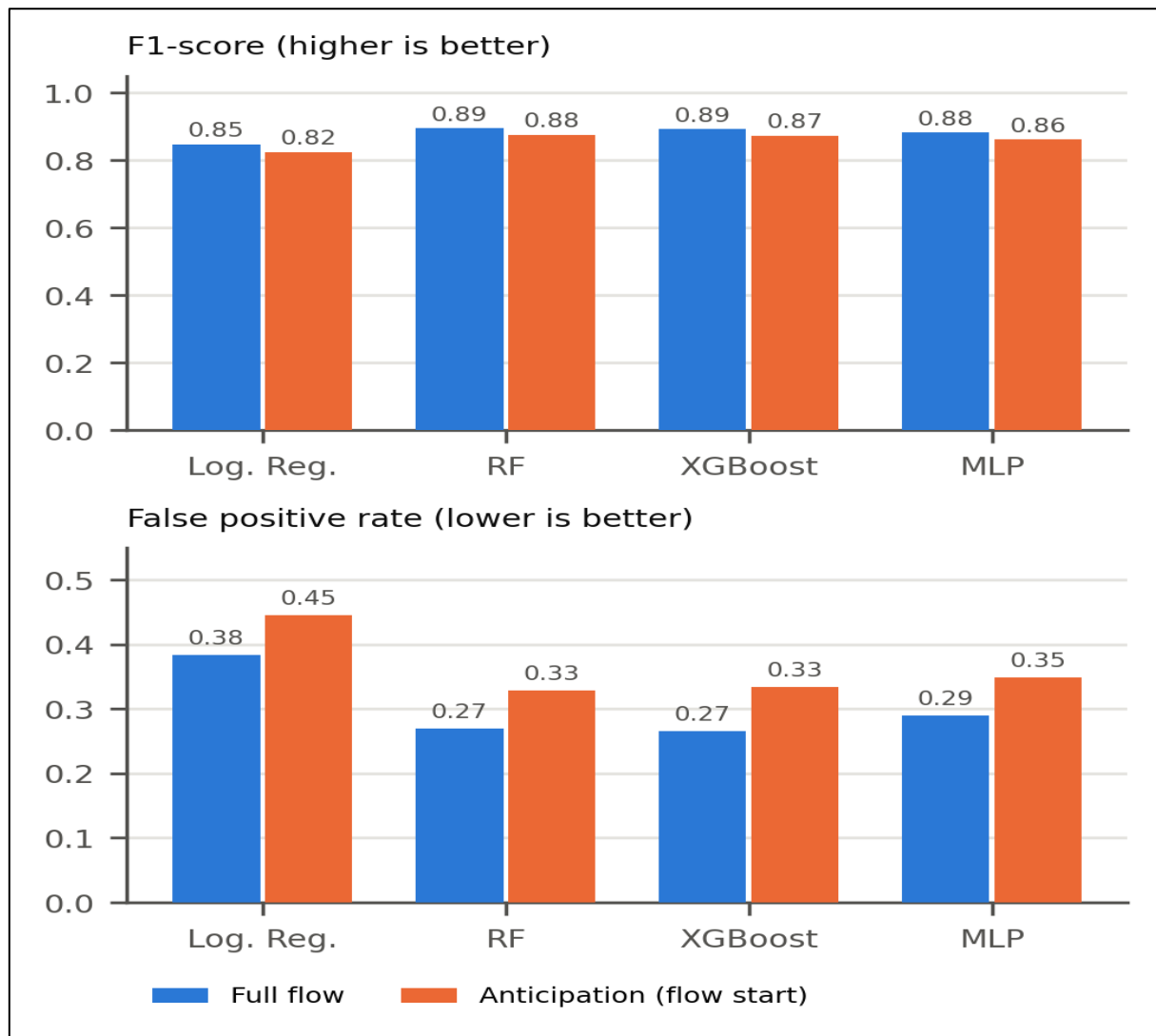


Fig 1 F1-Score and False Positive Rate Per Algorithm in the Detection and Anticipation Scenarios.

➤ Anticipation from the Start of the Connection

Removing the 22 features that are only known at the end of the flow had a moderate impact. For XGBoost, the AUC dropped from 0.982 to 0.978 and the F1-score from 0.892 to 0.873; recall remained practically unchanged (98.5%), but the FPR rose from 26.5% to 33.5%. The behavior was similar for the other algorithms (Table II). The anticipation model therefore still recognizes almost all attacks, but with more false

alarms, because it has less information to confirm that a connection is legitimate.

Fig. 2 shows the XGBoost ROC curves for both scenarios. The curves almost overlap for FPR above 10% and diverge only in the very low FPR region, where the full model keeps an advantage.

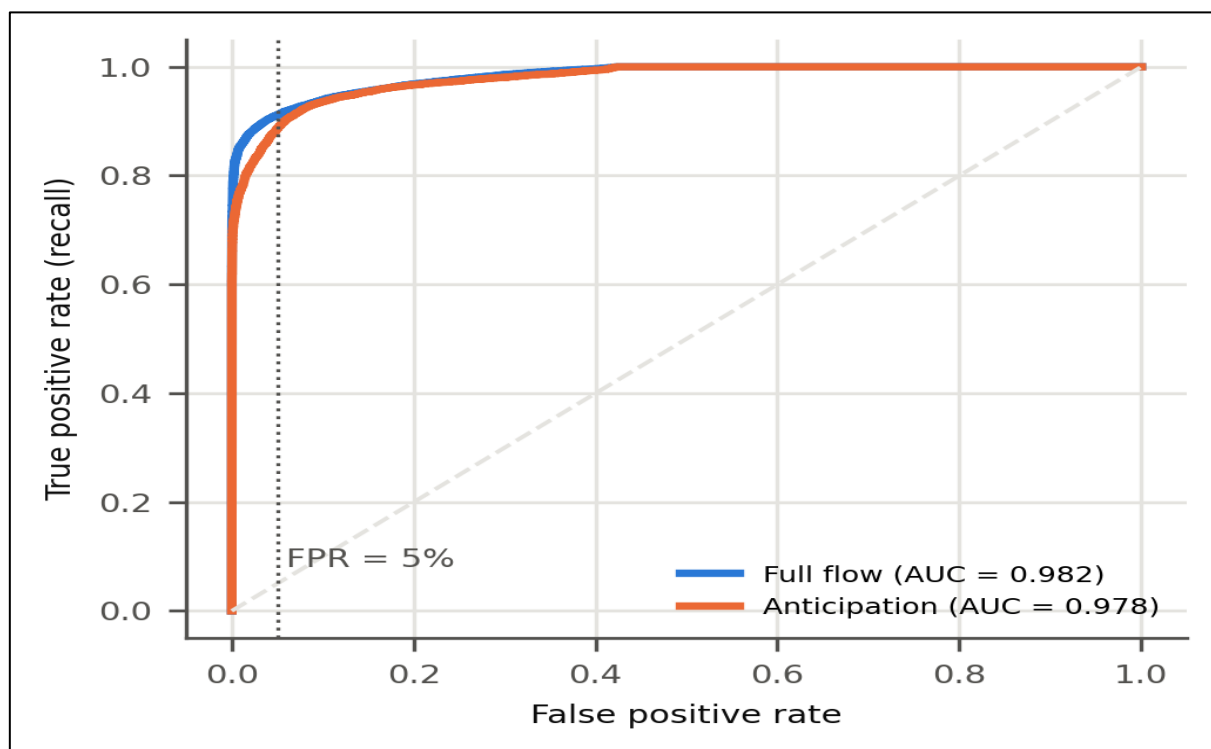


Fig 2 XGBoost ROC Curves in the Detection and Anticipation Scenarios.

The per-category analysis (Fig. 3) shows that the anticipation model detects at least 99.6% of the Reconnaissance, DoS, Exploits, Backdoor, Shellcode, Generic and Worms attacks. The Reconnaissance category deserves particular attention because it corresponds to the reconnaissance phase of the *kill chain*, which precedes exploitation: detecting it with 99.7% sensitivity right at connection start is, in practice, anticipating an attack that is still

being prepared. The most difficult category is Fuzzers (86.8% in the detection scenario and 90.3% in the anticipation scenario), whose traffic, made of random data sent to legitimate services, resembles normal traffic. For Analysis and Fuzzers the anticipation model even outperformed the full model, which is explained by its greater tendency to classify connections as attacks, the same tendency that raises its FPR.

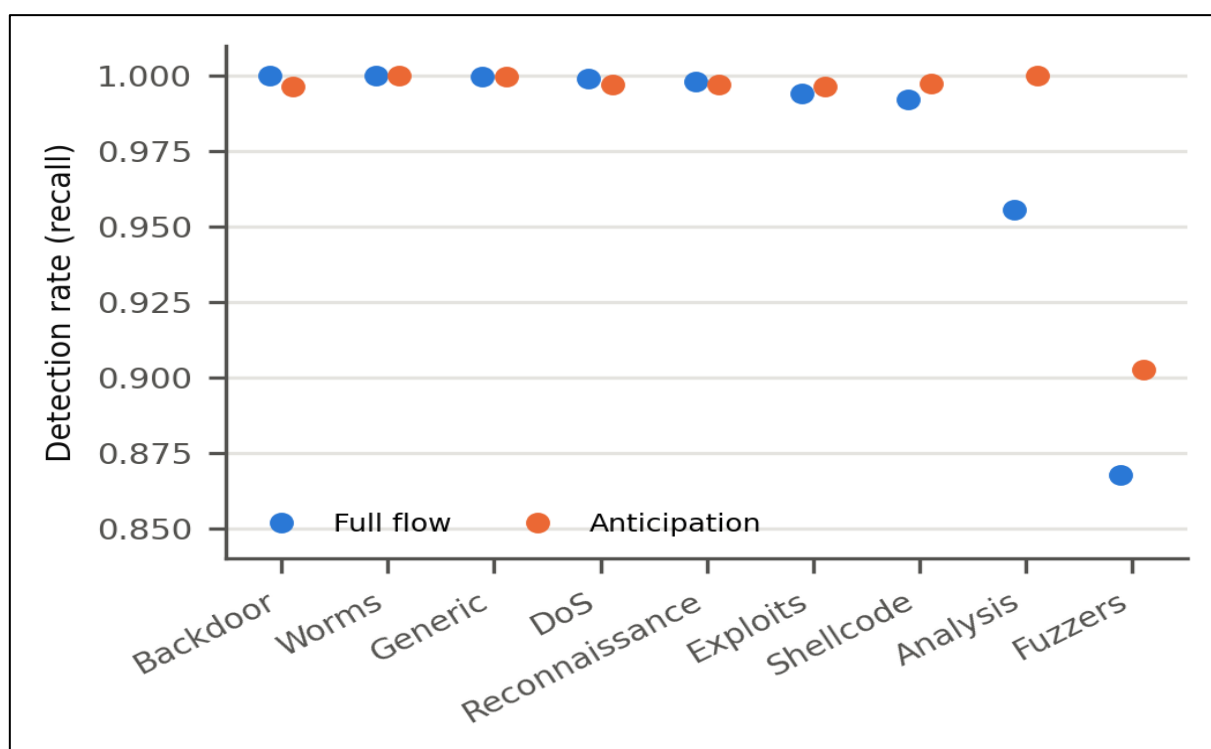


Fig 3 Detection Rate Per Attack Category (XGBoost).

➤ Decision Threshold Tuning

The classifiers output an attack probability, and the 0.5 threshold is merely a convention. By moving along the ROC curve, it is possible to choose the operating point that best serves the institution. Table 3 shows the XGBoost detection rate for three FPR levels. Accepting 5% false alarms, the full

model detects 91.1% of attacks and the anticipation model 88.6%; with 1% false alarms, the values are 85.5% and 77.5%. These operating points were read from the ROC curve of the test set and describe the achievable trade-off. In a real deployment, the threshold should be set using validation traffic from the network itself.

Table 3 XGBoost Detection Rate for Different False Positive Levels

False positive rate	Detection (full flow)	Anticipation (flow start)
1%	85.5%	77.5%
5%	91.1%	88.6%
10%	93.9%	93.6%

➤ Most Relevant Features and Robustness Test

Fig. 4 shows the 15 most important features in XGBoost. Source TTL (sttl) clearly dominates, with 84.3% of the total gain, followed by context counters (ct_srv_dst,

ct_dst_sport_ltm) and connection state. In the anticipation scenario, sttl remains at the top and the source TCP window (swin) gains relevant weight (14.4%).

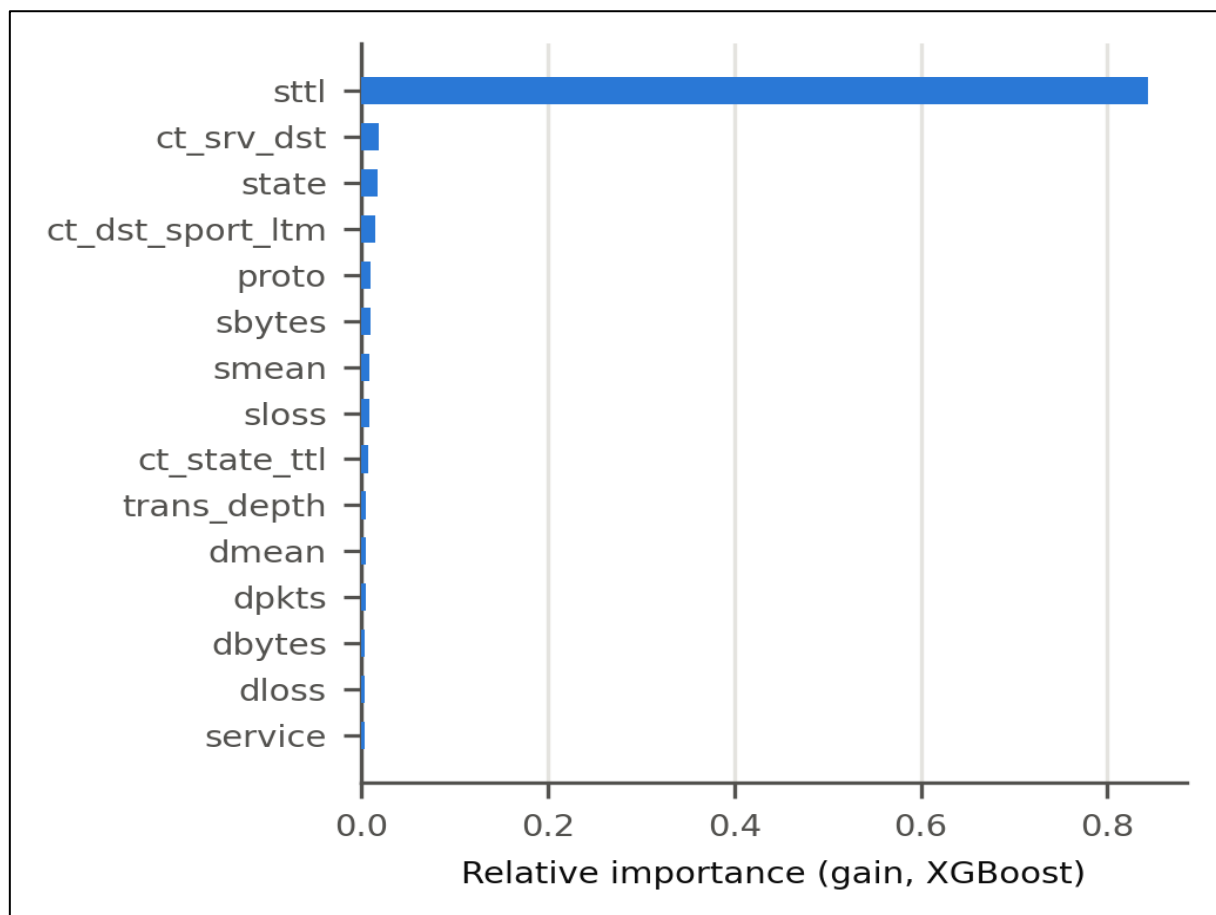


Fig 4 Relative Importance of the 15 Most Relevant Features (XGBoost, Detection Scenario).

The excessive weight of TTL calls for caution. In UNSW-NB15, the attacks were generated by equipment whose TTL values differ from those of normal traffic, so this feature may reflect a characteristic of the laboratory rather than of the attack itself, precisely the kind of spurious feature that Arp et al. [12] warn about. To test the robustness of the models, XGBoost was retrained without the three TTL-derived features (sttl, dttl and ct_state_ttl). In the detection scenario, performance remained practically unchanged (F1-score of 0.890, AUC of 0.981 and FPR of 26.8%), showing that redundant information in the remaining features compensates for the absence of TTL. In the

anticipation scenario, the loss was larger (F1-score of 0.849, AUC of 0.954 and FPR of 35.5%), because with fewer features the model relies more on TTL. This is the most conservative result and probably the closest to what would be obtained on a real network.

➤ Attack Type Identification

The multiclass classifier achieved an overall accuracy of 76.0% and a macro-averaged F1-score of 0.510. Fig. 5 shows the normalized confusion matrix. Categories with a distinctive traffic signature were well identified: Generic (F1-score of

0.984), Reconnaissance (0.864) and Normal (0.845). In contrast, Analysis, Backdoor and DoS were mostly confused with Exploits, with F1-scores of 0.099, 0.040 and 0.202, respectively. This confusion, also reported in the literature (the multiclass accuracy of 76.0% is comparable to the 75.6%

reported in [19]), is due to the similarity of the flows in these categories and to the small number of examples in the minority classes. In practice, the system knows it is facing an attack, but does not always identify its type correctly.

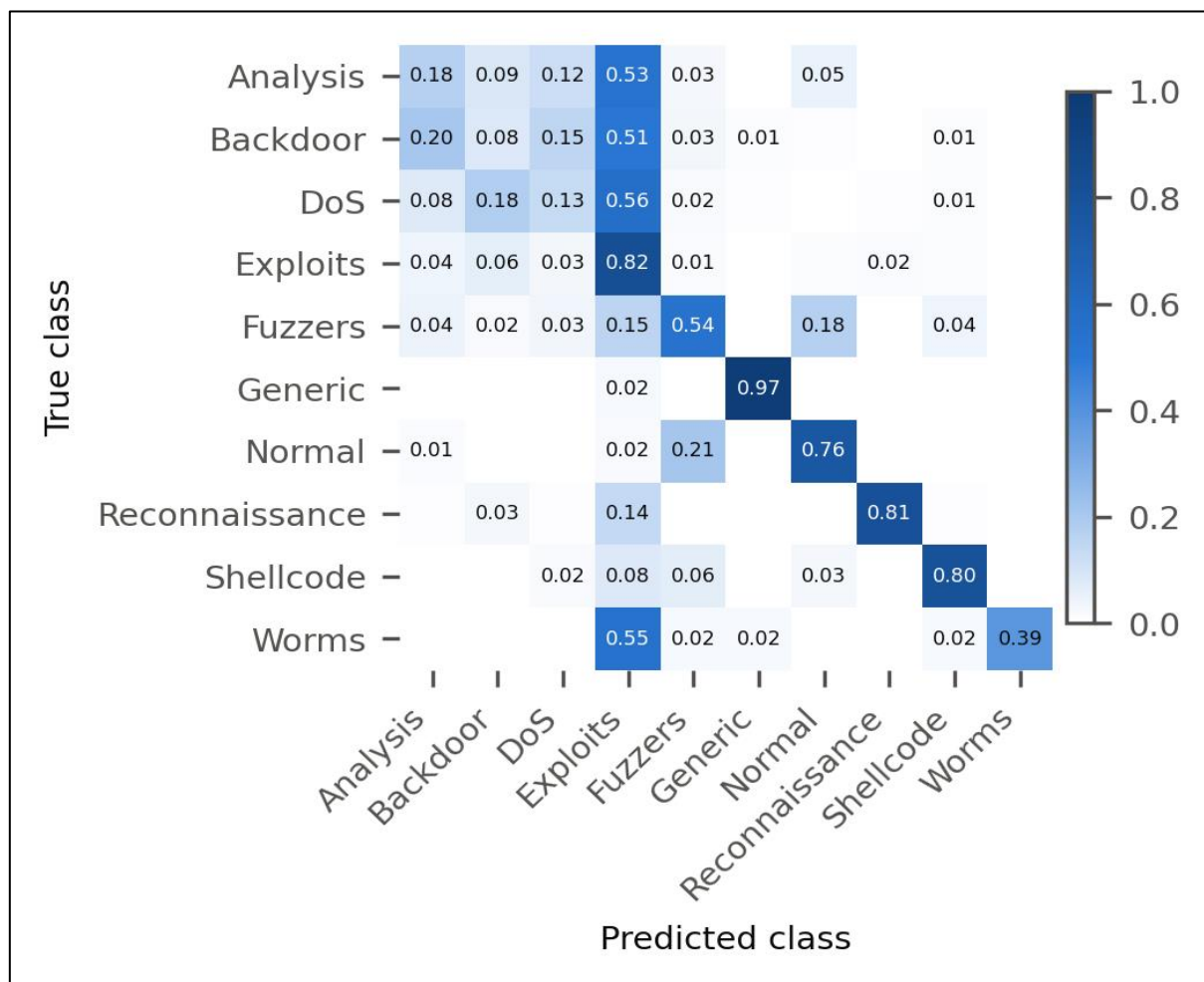


Fig 5 Normalized Confusion Matrix of the Multiclass Classification (XGBoost).

V. DISCUSSION

The results answer the research problem affirmatively, albeit with caveats. The best models detected about 98% of the attacks on an independent test set, and most of that capability was retained when the model only had access to connection-start information. The loss associated with anticipation shows up mainly as an increase in false alarms rather than a reduction in detection. For an institution, this means it is possible to act earlier, provided that more alerts are reviewed or that the early decision is combined with a confirmation at the end of the flow.

The superiority of tree-based methods confirms the trend reported in the literature for tabular data [5], [4]. Among them, XGBoost offers the best balance between performance, training speed and inference cost, making it the most suitable candidate for institutions with limited computing resources. The MLP did not compensate for its greater complexity.

The FPR at the default threshold is the main operational obstacle. In a typical institutional network the vast majority of

flows are legitimate, and an FPR of 27% would generate an unmanageable volume of alerts. Threshold tuning substantially reduces the problem, as shown in Table III, but the underlying lesson, also stressed by Sommer and Paxson [11], is that a model trained in a laboratory cannot be deployed without calibration on the organization's own traffic. The distribution shift between the UNSW-NB15 training and test partitions already anticipates, on a small scale, what will happen when the model meets a different network.

The test without TTL features showed that the detection model does not depend on a single dataset artifact, which strengthens the validity of the results. The anticipation model is more dependent on it, and the values obtained without TTL (AUC of 0.954) should be taken as the most prudent estimate of its performance.

For Mozambican institutional networks, including those of military education institutions, the results suggest a three-layer architecture. The first layer is a sensor that extracts flows from traffic using open-source tools such as Zeek or Argus, the

latter having been used to build UNSW-NB15. The second is an XGBoost anticipation model that scores each connection at its start and applies a conservative threshold. The third is a full-flow detection model that confirms or dismisses the alert before it reaches the analyst. This architecture complements the access control mechanisms of a Zero Trust model by providing them with a continuous signal about connection behavior, and can feed the sectoral and national CSIRTs foreseen in the strategy under preparation [1].

VI. CONCLUSION

This paper evaluated four machine learning algorithms for the detection and anticipation of cyberattacks using the UNSW-NB15 dataset. XGBoost and Random Forest obtained the best results, with an AUC close to 0.98 and detection of more than 97% of attacks. Using only the 20 features available at connection start, XGBoost retained an AUC of 0.978, demonstrating the feasibility of anticipation. The false positive rate is the main limitation, but it can be controlled by tuning the decision threshold, which allows about 91% of attacks to be detected with 5% false alarms.

The study has limitations that should be taken into account. The data come from a laboratory environment rather than a Mozambican network, so the absolute values are not directly transferable. The UNSW-NB15 partitions do not preserve the temporal order of flows, which prevented modeling anticipation as the prediction of event sequences. The hyperparameters were not exhaustively optimized, and the operating points in Table III were read on the test set itself.

As future work, we propose to (i) collect and label traffic from a national institutional network, with the appropriate authorizations, to train and validate the models under real conditions; (ii) use the full UNSW-NB15 or CICIDS2017 files, which preserve temporal order, to test sequential models such as LSTM networks for predicting multi-stage attacks; (iii) apply explainability techniques, such as SHAP values, so that analysts can understand each alert; and (iv) test the proposed architecture in a pilot environment integrated with the institution's access control mechanisms.

REFERENCES

- [1]. L. Khoza, "Mozambique fast-tracks AI cyber strategy," *ITWeb Africa*, Jun. 22, 2026. [Online]. Available: <https://itweb.africa/article/mozambique-fast-tracks-ai-cyber-strategy/xA9PO7NEVbdvo4J8>
- [2]. M. J. Z. de Barros and B. Van Niekerk, "Cybersecurity in Mozambique: Status and challenges," in *Proc. 22nd Eur. Conf. Cyber Warfare Secur. (ECCWS)*, 2023, pp. 553–558.
- [3]. e-Governance Academy, "National Cyber Security Index: Mozambique," 2026. [Online]. Available: <https://ncsi.ega.ee/country/mz/>
- [4]. A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1153–1176, 2016, doi: 10.1109/COMST.2015.2494502.
- [5]. Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 1, Art. no. e4150, 2021, doi: 10.1002/ett.4150.
- [6]. M. Husák, J. Komárková, E. Bou-Harb, and P. Čeleda, "Survey of attack projection, prediction, and forecasting in cyber security," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 640–660, 2019, doi: 10.1109/COMST.2018.2871866.
- [7]. E. M. Hutchins, M. J. Cloppert, and R. M. Amin, "Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains," *Leading Issues Inf. Warfare Secur. Res.*, vol. 1, no. 1, pp. 80–106, 2011.
- [8]. D. E. Denning, "An intrusion-detection model," *IEEE Trans. Softw. Eng.*, vol. SE-13, no. 2, pp. 222–232, Feb. 1987, doi: 10.1109/TSE.1987.232894.
- [9]. L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.
- [10]. T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.
- [11]. R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *Proc. IEEE Symp. Secur. Privacy*, 2010, pp. 305–316, doi: 10.1109/SP.2010.25.
- [12]. D. Arp et al., "Dos and don'ts of machine learning in computer security," in *Proc. 31st USENIX Secur. Symp.*, 2022, pp. 3971–3988.
- [13]. M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl. (CISDA)*, 2009, pp. 1–6, doi: 10.1109/CISDA.2009.5356528.
- [14]. I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy (ICISSP)*, 2018, pp. 108–116, doi: 10.5220/0006639801080116.
- [15]. G. Engelen, V. Rimmer, and W. Joosen, "Troubleshooting an intrusion detection dataset: The CICIDS2017 case study," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, 2021, pp. 7–12, doi: 10.1109/SPW53761.2021.00009.
- [16]. N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, Canberra, Australia, 2015, pp. 1–6, doi: 10.1109/MilCIS.2015.7348942.
- [17]. N. Moustafa and J. Slay, "The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Inf. Secur. J.: Global Perspective*, vol. 25, no. 1–3, pp. 18–31, 2016, doi: 10.1080/19393555.2015.1125974.
- [18]. M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion

- detection data sets,” *Comput. Secur.*, vol. 86, pp. 147–167, 2019, doi: 10.1016/j.cose.2019.06.005.
- [19]. S. M. Kasongo and Y. Sun, “Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset,” *J. Big Data*, vol. 7, Art. no. 105, 2020, doi: 10.1186/s40537-020-00379-6.
- [20]. F. Pedregosa *et al.*, “Scikit-learn: Machine learning in Python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.