Evaluating Random Forest Performance on Packet- and Flow-Level Features for Network Traffic Classification

MACHIEL LUNING, University of Twente, The Netherlands

This paper evaluates the performance of Random Forest models for network traffic classification using two different feature sets, namely packet- and flow-level features. These features are extracted from an existing Internet traffic capture and label the data using histogram-based methods. Two distinct models are trained, one on packet-level features and the other on flow-level features. The performance of the models is assessed based on accuracy, precision, and recall, and a feature importance analysis is conducted. The results show the potential of Random Forest for effective network traffic classification and provide insights into the importance of packet- and flow-level features for such tasks.

Additional Key Words and Phrases: Machine Learning (ML), Random Forest, Network Traffic Classification, Packet-Level Features, Flow-Level Features, Feature Importance

1 INTRODUCTION

An ever-increasing number of people make use of the Internet. In 2024, a total of 5.5 billion people were online, or around 68% of the world's population [2]. All these people produce a wide range of types of network traffic, and Internet Service providers need to identify these types to manage the performance or ensure the security of the network. Traditional classification methods, such as port-based or Deep Packet Inspection, fail due to multiple reasons which will be in the related work section. However, machine learning-based approaches offer a promising and suitable alternative to these traditional methods [1, 4, 5].

This paper aims to evaluate and compare the performance of Random Forest, based on accuracy, precision, and recall, for network traffic classification using packet-level and flow-level features. Packet-level features represent individual packet characteristics, like size and inter-arrival time, while flow-level features are the combined information of multiple packets. By comparing the performance of the two models, this paper aims to provide insight into the usefulness of each feature set for network classification using Random Forest.

Furthermore, this paper will also conduct a feature importance analysis for both models to assess the necessity of each feature in their respective feature sets.

1.1 Research Question

RQ1 How does the performance of Random Forest models trained on packet- versus flow-level features compare for network traffic classification?

RQ2 How necessary is each feature in its corresponding feature set?

2 RELATED WORK

Network traffic classification is a topic that has evolved significantly over the years. The earliest approach to network classification was based on ports. These ports were defined by the Internet Assigned Numbers Authority and coupled to an application [1, 4, 5]. However, this method became ineffective due to applications nowadays using unregistered or randomly generated ports [1, 3–5].

Deep Packet Inspection (DPI) emerged as an alternative to port-based classification. Instead of relying on ports, this technique analyzes the packets' payload. A signature is extracted from this payload and then matched to an already existing library of predefined signatures. Although it overcomes the problems of port-based classification, this technique has its own problems. Firstly, DPI is not able to classify encrypted traffic. This impacts its performance significantly, since many applications encrypt their data [1, 4, 5]. Furthermore, it raises privacy concerns, since accessing packet content is a breach of privacy policy and law in different countries [1, 4].

Machine learning emerged as a solution that could solve the problems of both previous techniques. This is because machine learning techniques do not solely rely on port numbers or access the packets' payload. Instead, it relies on statistical features of network packets or flows to classify. These include features such as packet size or inter-arrival time for single packets or duration and total size for flows [1, 4].

3 METHODOLOGY

3.1 Network traffic traces

For this research, network traffic traces from the University of New South Wales (UNSW) are used. These traces are made up of network traffic from 28 unique IoT devices as well as some non-IoT devices, such as phones and laptops. The IoT devices consist of cameras, switches and triggers, hubs, air quality sensors, electronics, healthcare devices, and light bulbs. A full list of the used devices can be found in Appendix A. This data was originally captured over a period of 26 weeks, however, only two weeks of data was made public. The data contains approximately 11,5 GB of raw packet captures and is available at: https://iotanalytics.unsw.edu.au/iottraces.html [6].

Two raw packet capture files, from September 24th and 28th, are not used due to issues encountered during feature extraction. These issues caused the feature extraction to stop midway and, thus, be incomplete. Multiple debugging efforts failed to resolve these issues, so the decision was made to exclude these files.

3.2 Feature Extraction

Feature extraction is a necessary step in transforming raw packet captures into a usable form for training a machine learning model. This section describes the extraction of packet-level and flow-level features used in this research.

TScIT 42, January 31, 2025, Enschede, The Netherlands

3.2.1 Packet-level features. Packet-level features are characteristics of individual packets in the captured traffic. The following packet-level features were extracted:

- Packet size
- Protocol
- Source IP
- Destination IP
- Source port
- Destination port
- Inter-arrival time

These features were extracted using Python making use of the PyShark package. The script processes a raw packet capture file by iterating through each packet and extracting the features. The inter-arrival time is extracted from the time_delta field of the transport layer.

3.2.2 Flow-level features. Flow-level features combine the information from multiple packets that belong to the same flow. For this research, bidirectional flows are used, which contain packets sent in both directions between the two endpoints. The following flow-level features were extracted:

- Source IP
- Destination IP
- Source port
- Destination port
- Protocol
- Flow volume
- Flow duration
- Flow rate
- Packet count

The flow-level features were calculated using Python and the packet-level features. The flows were identified using the five-tuple (source IP and port, destination IP and port, and protocol) and the inter-arrival time, which is zero for the first packet of a new flow.

3.3 Data Labeling

In supervised machine learning, which Random Forest is part of, labeling the dataset is required in order to train the model.

The labels given to the packet-level dataset include labels for packet size and inter-arrival time, as well as a device pair label. The latter shows from which type of device the packet came and what kind of device its destination was.

The labels for the flow-level dataset include labels for flow volume, flow duration, average flow rate, packet count, and again a device pair label. However, the meaning of the device pair label is a bit different. Since the flows are bidirectional, the device pair does not represent a direction.

Histograms were used for all labels, except the device pair label. These histograms were created using Python and can be found in Appendix B. By analyzing the distribution, distinct ranges were identified. For some labels, logarithmic bins were used due to skewness toward lower values.

The following ranges are identified:

- **Packet size:** small (<= 200 bytes), medium (<=1200 bytes), and large (1200+ bytes)
- **Inter-arrival time:** very short (<= 1 second), short (<= 10 seconds), moderate (<= 40 seconds), and long (40+ seconds)
- **Flow volume:** Very low (<= 1000 bytes), low (<= 5000 bytes), medium (<= 10000 bytes), and high (10000+ bytes)
- **Flow Duration:** short (<= 100 seconds), medium (<= 1000 seconds), and long (1000+ seconds)
- **Flow rate:** very slow (<= 15000 B/s), slow (<= 140000 B/s), medium (<= 440000 B/s), and fast (440000+ B/s)
- **Packet count:** single packet (1 packet), low (<= 5 packets), medium (<= 21 packets), and high (21+ packets)

For the device pair label, the list of devices provided by the study from the UNSW [6] was used. Apart from the names of the devices, the list also contains the MAC addresses of these devices. Using this, it was possible to determine what kind of device the source and destination devices of a packet or flow were. The category of each device can be found in Appendix A.

3.4 Model Development

To evaluate the performance of Random Forest for network traffic classification, two models were developed: one trained on packet-level features and the other on flow-level features. Both models were implemented using the Random Forest classifier from the Python package Scikit-learn with its default hyperparameters. The decision to use the default parameters was mainly due to limited time, however, this should still allow for a meaningful performance comparison between the datasets. These settings include 100 estimators and a maximum depth determined by the model itself. For both models a train-test split of 80-20 was used.

4 RESULTS

4.1 Packet-Level Model

Actual value			
Small	2009183	0	0
Medium	0	349634	0
Large	0	0	895087
	Small	Medium	Large
	Predicted value		

Table 1. Confusion Matrix for packet size label

Actual value				
Very Short	2944037	0	0	0
Short	0	135745	0	0
Moderate	0	0	125502	1
Long	0	0	0	48619
	Very Short	Short	Moderate	Long
	Predicted value			

Table 2. Confusion Matrix for inter-arrival time label

The packet-level model performs well across its outputs, for all its labels, accuracy, precision, and recall were calculated using Python.

This can also be done with the confusion matrices in Table 1 and 2, and Appendix C.1. Accuracy measures overall correctness, precision measures the proportion of correct predictions, and recall measures the proportion of actual values that were correctly predicted.

For the packet size label, the model achieved perfect scores, with accuracy, precision, and recall of 1.0. For the inter-arrival time label, it achieved similar results, with an accuracy of 0.9999997, a precision of 0.9999949, and a recall of 0.9999980. Since these labels are directly based on distinct ranges of their corresponding features, these results can be expected.

The device pair label is not directly tied to one distinct feature and the model's performance for this feature also reflects this. However, it still has a strong performance. It scored an accuracy of 0.9906549, a precision of 0.9012846, and a recall of 0.9014629.

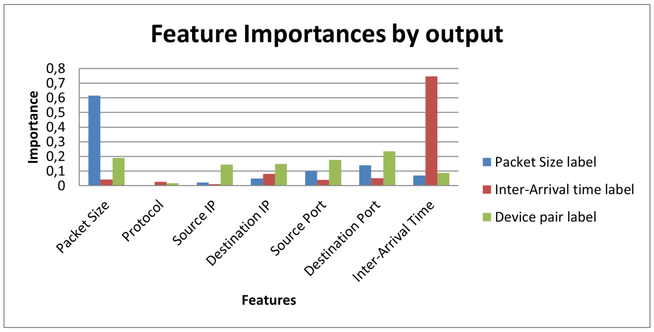

Fig. 1. Feature importance packet-level model

The feature importance analysis, shown in figure 1, reveals the contribution of each packet-level feature to each label. For both the packet size and inter-arrival time labels, the corresponding feature is of the most importance, which makes sense since they were labeled using those features. The fact that the device pair label is not directly linked to a single feature is reflected in the analysis. For this label, almost all features have relatively similar importance, with the destination port being slightly more important, the inter-arrival time slightly less important, and protocol being the clear exception.

4.2 Flow-level Model

Actual value				
Very Low	59605	0	0	0
Low	0	13336	0	0
Medium	0	2	5860	1
High	0	0	0	2905
	Very Low	Low	Medium	High
	Predicted value			

Table 3. Confusion Matrix for flow volume label

Actual value			
Short	77176	1	0
Medium	0	2797	0
Long	0	0	1734
	Short	Medium	Long
	Predicted value		

Table 4. Confusion Matrix for flow duration

Actual value				
Very Slow	47605	0	0	0
Slow	0	21776	0	0
Medium	0	0	9714	0
Fast	0	0	0	2613
	Very Slow	Slow	Medium	Fast
	Predicted value			

Table 5. Confusion Matrix for flow rate label

Actual value				
Single Packet	26971	0	0	0
Low	0	18970	0	0
Medium	0	0	29823	0
High	0	0	0	5944
	Single Packet	Low	Medium	High
	Predicted value			

Table 6. Confusion Matrix for packet count label

Same as the packet-level model, the flow-level model also performed strongly across its outputs. Like the packet-level model, for the labels that are based on distinct ranges of their corresponding feature, the model achieves almost perfect scores.

- **Flow volume label:** Accuracy: 0.9999755, Precision: 0.9999625, Recall: 0.9999147
- **Flow duration label:** Accuracy: 0.9999878, Precision: 0.9998809, Recall: 0.9999957
- **Flow rate and packet count labels:** For both labels, the model achieved perfect scores, with accuracy, precision, and recall of 1.0

On the device pair label, the flow-level model also achieved a lower but still strong performance. It scored an accuracy of 0.9674695, a precision of 0.9545989, and a recall of 0.9276326.

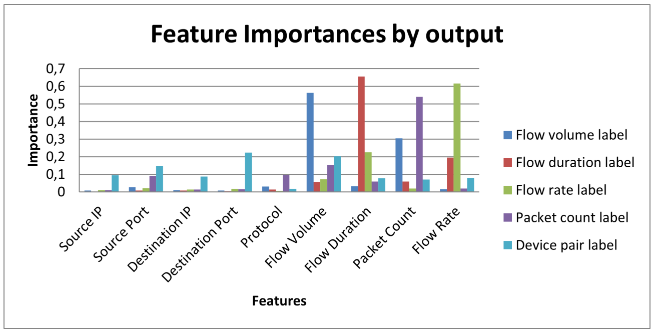

Fig. 2. Feature importance flow-level model

The feature importance analysis for the flow-level model, shown in figure 2, again shows that the labels labeled using a single feature rely mostly on that feature. However, this is to a lesser extent than compared to the packet-level model. It can be seen that these labels also significantly rely on other features as well. The device pair label, like in the packet-level model, again relies on almost all features in varying degrees, with the protocol being the clear exception for this model as well.

5 DISCUSSION

5.1 Research question 1

From the results, it can be seen that both models perform very well. They both achieved perfect or almost perfect scores for all labels based on distinct ranges of their corresponding features. For the device pair label, the flow-level model achieved a lower accuracy than the packet-level model, however, it did achieve a higher precision and recall.

If classification were performed solely using the labels based on ranges, for example by creating profiles for different applications, one model would not necessarily outperform the other since they both score almost perfectly for these labels. However, since the flow-level model has more of these labels, it might be possible to create more fine-grained profiles for this model.

Which model performs better if the device pair label were to be used depends on the objective of the network traffic classification. For goals like resource management or detecting malicious traffic, the flow-level model would perform better due to its higher precision and recall. A lower precision and recall could result in wasted resources by allocating, for example, bandwidth to an application that does not require it, or result in missing malicious traffic or flagging normal traffic as malicious. However, if overall classification is the goal, the packet-level model would outperform the flow-level model due to its higher accuracy.

5.2 Research question 2

From the feature importance analyses, it can be concluded that the protocol could be excluded from the packet-level features with minimal impact since all labels only marginally rely on it. However, it is not possible to conclude this for the flow-level features as well since the packet count label has a significant reliance on it.

For both feature sets, none of the other features can be excluded without significant impact. Mainly the device pair label relies on most features in various but non-discardable amounts, with the exception being the protocol. If this label were not to be used, more features could be excluded without having a significant impact on the remaining labels.

6 CONCLUSION

This research highlights the potential of Random Forest models for network traffic classification using both packet- and flow-level features. The results show the strengths of each feature set in different contexts. The packet-level model is better suited than the flow-level model for overall classification due to its higher accuracy, while the flow-level model given its higher precision and recall would outperform the packet-level model in tasks requiring a more nuanced insight, such as resource allocation.

The feature importance analyses also revealed the possible optimization that can be done in future work by excluding the protocol from the packet-level model without it having a significant impact.

In the end, this study not only shows the effectiveness of Random Forest in network classification, but it also provides insights for future research to refine feature selection.

7 LIMITATIONS

One possible limitation of this study is the lack of hyperparameter tuning during the development of the models. Both models were trained using default parameters, which may not be the optimal configuration for these datasets. The models achieved high accuracy, precision, and recall, especially for the labels based on distinct ranges of their corresponding features. This performance could be partially due to overfitting. Future work could evaluate both of the models' performances on entirely unseen datasets to provide further insight into this.

REFERENCES

[1] Ahmad Azab, Mahmoud Khasawneh, Saed Alrabaee, Kim Kwang Raymond Choo, and Maysa Sarsour. 2024. Network traffic classification: Techniques, datasets, and challenges. *Digital Communications and Networks* 10, 3 (6 2024), 676–692. https://doi.org/10.1016/J.DCAN.2022.09.009

[2] ITU. 2024. *Measuring digital development: Facts and Figures 2024*. Technical Report. International Telecommunication Union, Geneva. https://www.itu.int/hub/publication/D-IND-ICT_MDD-2024-4/

[3] Andrew W. Moore and Konstantina Papagiannaki. 2005. Toward the Accurate Identification of Network Applications. *Lecture Notes in Computer Science* 3431 (2005), 41–54. https://doi.org/10.1007/978-3-540-31966-5_4

[4] Fannia Pacheco, Ernesto Exposito, Mathieu Gineste, Cedric Baudoin, and Jose Aguilar. 2019. Towards the Deployment of Machine Learning Solutions in Network Traffic Classification: A Systematic Survey. *IEEE Communications Surveys and Tutorials* 21, 2 (4 2019), 1988–2014. https://doi.org/10.1109/COMST.2018.2883147

[5] Muhammad Shafiq, Xiangzhan Yu, Asif Ali Laghari, Lu Yao, Nabin Kumar Karn, and Foudil Abdessamia. 2017. Network Traffic Classification techniques and comparative analysis using Machine Learning algorithms. *2016 2nd IEEE International Conference on Computer and Communications, ICCC 2016 - Proceedings* (5 2017), 2451–2455. https://doi.org/10.1109/COMPCOMM.2016.7925139

[6] Arunan Sivanathan, Hassan Habibi Gharakheili, Franco Loi, Adam Radford, Chamith Wijenayake, Arun Vishwanath, and Vijay Sivaraman. 2019. Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics. *IEEE Transactions on Mobile Computing* 18, 8 (8 2019), 1745–1759. https://doi.org/10.1109/TMC.2018.2866249

A LIST OF DEVICES IN TRAFFIC CAPTURE

The following devices were used during the network traffic capture, as documented in the study from the University of New South Wales [6]. Added in parenthesis is the category in which they fall for the device pair label.

- Smart Things (Hub)
- Amazon Echo (Hub)
- Netatmo Welcome (Camera)
- TP-Link Day Night Cloud camera (Camera)
- Samsung SmartCam (Camera)
- Dropcam (Camera)
- Insteon Camera (Camera)
- Withings Smart Baby Monitor (Camera)
- Belkin Wemo switch (Switch or trigger)
- TP-Link Smart plug (Switch or trigger)
- iHome (Switch or trigger)
- Belkin wemo motion sensor (Switch or trigger)
- NEST Protect smoke alarm (Air quality sensor)
- Netatmo weather station (Air quality sensor)
- Withings Smart scale (Healthcare device)
- Blipcare Blood Pressure meter (Healthcare device)
- Withings Aura smart sleep sensor (Healthcare device)
- Light Bulbs LiFX Smart Bulb (Light bulb)
- Triby Speaker (Electronics)
- PIX-STAR Photo-frame (Electronics)
- HP Printer (Electronics)
- Samsung Galaxy Tab (Phone)
- Nest Dropcam (Camera)
- 2x Android Phone (Phone)
- Laptop (Laptop)
- 2x MacBook (Laptop)
- iPhone (Phone)
- TPLink Router Bridge LAN (Gateway)

B HISTOGRAMS

B.1 Packet-level model

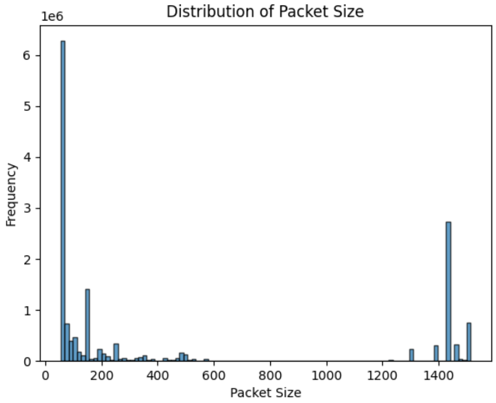

Fig. 3. Histogram for packet size

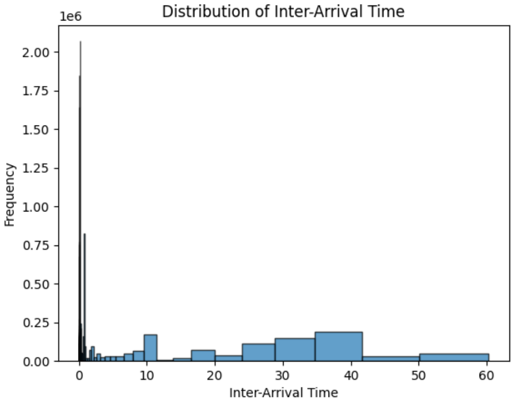

Fig. 4. Histogram for inter-arrival time (logarithmic bins)

B.2 Flow-level model

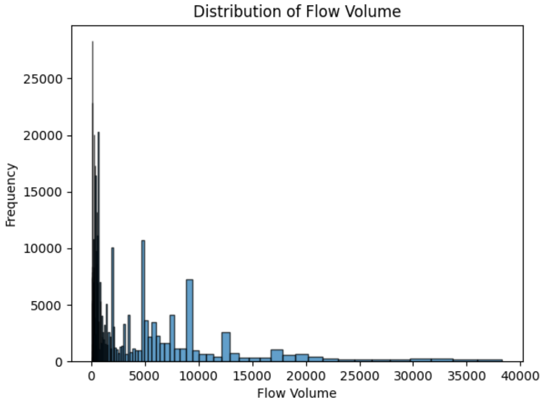

Fig. 5. Histogram for flow volume (logarithmic bins)

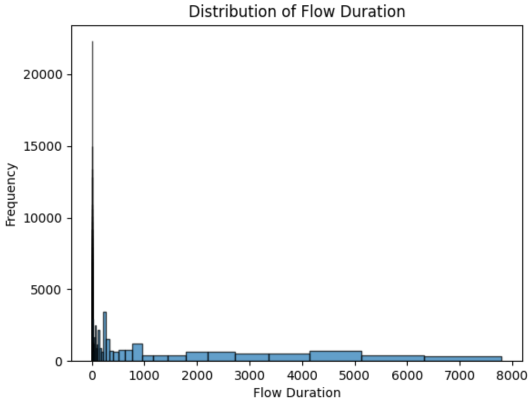

Fig. 6. Histogram for flow duration (logarithmic bins)

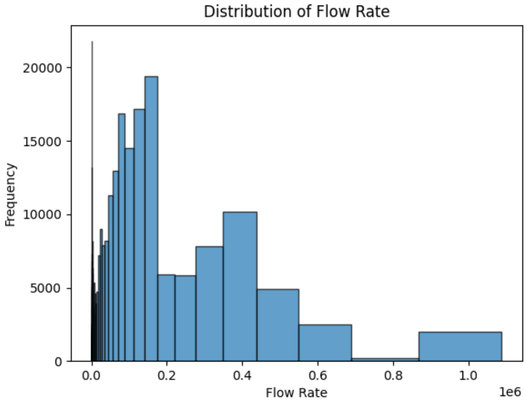

Fig. 7. Histogram for flow rate (logarithmic bins)

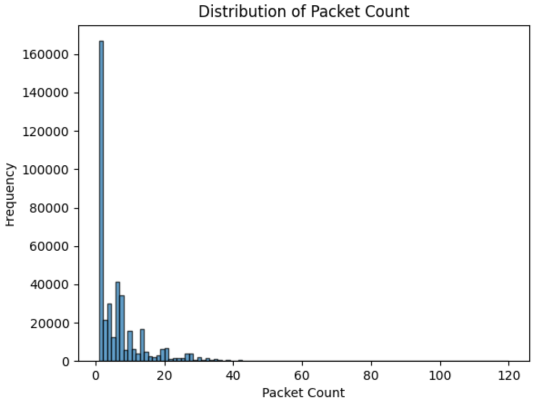

Fig. 8. Histogram for packet count

C CONFUSION MATRICES DEVICE PAIRS

C.1 Packet-Level Model

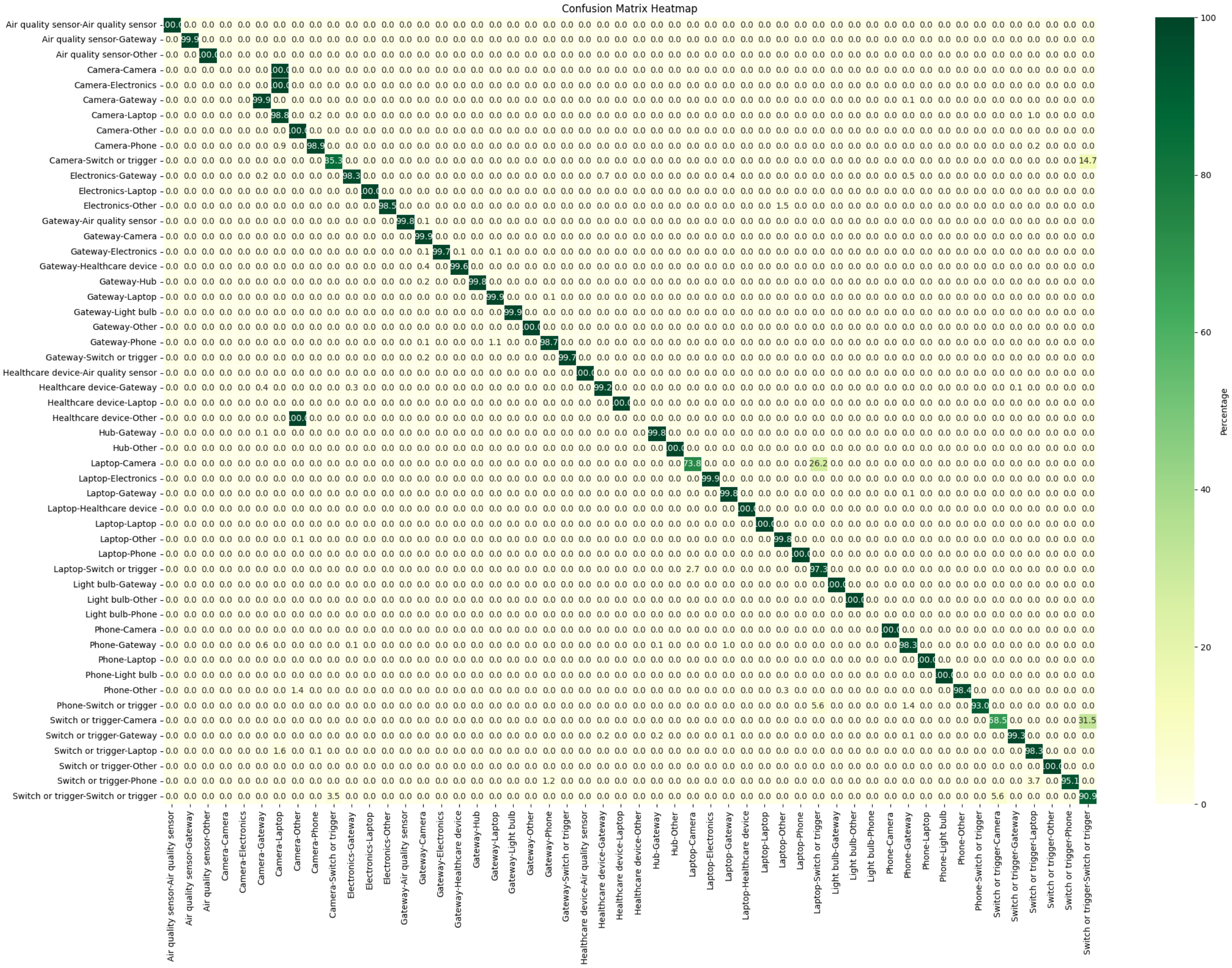

Fig. 9. Confusion matrix for device pair label (packet-level model)

C.2 Flow-Level Model

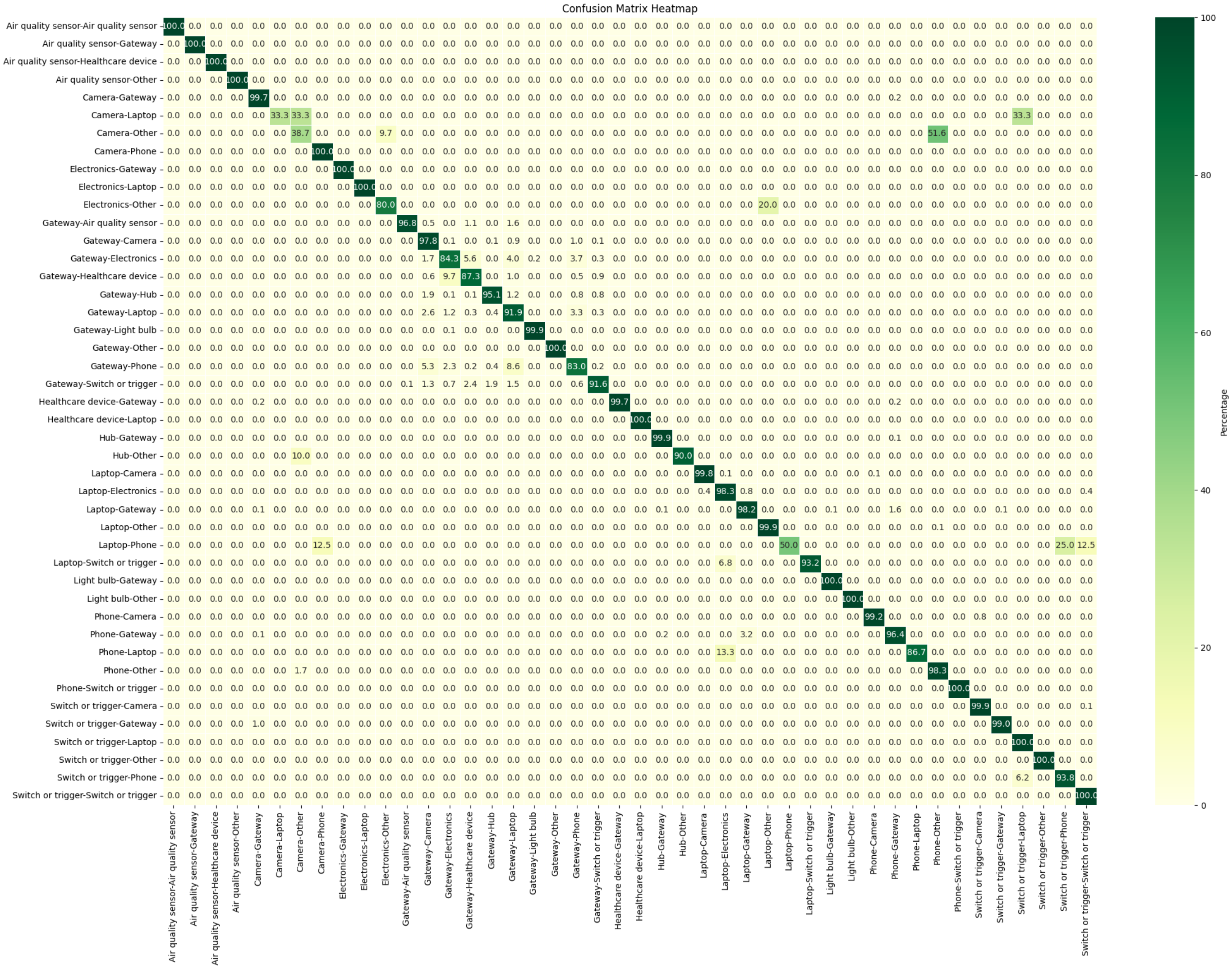

Fig. 10. Confusion matrix for device pair label (flow-level model)