# Exploring the Reliability of IoT Packet Classifiers: An Experimental Study

Aleksandar Pasquini*, Rajesh Vasa*, Hassan Habibi Gharakheili†, Irini Logothetis*,
Minh Tran‡ and Alexander Chambers‡
* A2I2, Deakin University, Geelong, Australia. {*aleksandar.pasquini, rena.logothetis, rajesh.vasa*}*@deakin.edu.au*
† School of EE&T, UNSW, Sydney, Australia. *h.habibi@unsw.edu.au*
‡ Information Sciences Division, Defence Science & Technology Group, Edinburgh SA, Australia.
{*minh.tran7, alexander.chambers*}*@defence.gov.au*

***Abstract*—Making robust inferences from IoT network traffic is needed for the reliable management of IoT devices in health, industrial, or agricultural domains, particularly at scale. Ensemble methods aim to enhance accuracy and consistency by aggregating predictions from multiple classification models. However, the process of combining several models can be difficult due to disagreements in their predictions. This research examines the causes of packet-based classifiers disagreement, specifically when and why packets receive multiple prediction labels. We develop a method that identifies packets that cause disagreement and show how disagreements can provide information on the reliability of the prediction from the classifier. For this study, we adapted three classifiers from previous research on IoT traffic inference. The classifiers were applied to public traces of IoT packets. Our results indicate that disagreement varies across IoT classes and is lower in certain protocols, such as SSDP and NTP at the application layer and LLC at the datalink layer. We also found that when the three classifiers agree on a prediction, there is a 97% chance that it is correct. By removing disagreeing predictions, an ensemble classifier's accuracy can be increased.**

***Index Terms*—Machine Learning, Packet Classification, Reliable Inference, Internet of Things**

## I. Introduction

As the Internet of Things (IoT) continues to expand, the need for accurate device characterization becomes paramount. IoT devices vary in their functionalities, characteristics, and behavior, making it challenging to develop comprehensive models that capture their diverse nature. Ensemble techniques have emerged as a promising approach to address this issue by combining multiple models to improve the accuracy of device characterization. By leveraging the diversity of individual models, ensemble methods can capture a broader range of device behaviors, leading to more robust and reliable characterizations. Such ensemble techniques have shown remarkable success in various domains, including anomaly detection, classification, and predictive maintenance in IoT environments [1]–[3].

An intriguing aspect arises when these ensemble learners produce conflicting or disagreeing predictions. This phenomenon invites exploration into the reasons behind such disparities. Unraveling the underlying factors driving these disagreements can offer valuable insights into the strengths and limitations of ensemble methods, shed light on the inherent complexities of the data, and potentially pave the way for novel techniques to enhance ensemble learning performance. However, current metrics do not explicitly leverage this information. We propose a new metric called "Disagreement" and exploit it to improve the overall ensemble accuracy.

We propose three categories of disagreement. The first category is **No Disagreement**, which occurs when all classifiers provide the exact same label prediction for a given input instance. This label may or may not be the same as the ground truth label. The second category is **Partial Disagreement**, which arises when the majority of classifiers agree on a particular label for the input instance. Finally, the third category is **Full Disagreement**, which happens when none of the classifiers share the same predicted label for the input instance.

In this paper, we ensemble three recent state-of-the-art packet-based classifiers [4]–[6]. We then group the data by the number of labels the packets receive and analyze if the labels differ from the ground truth. This study focuses on packet-based classifiers; however, this method can be modified to work with other types of classifiers, such as flow-based.

The objective of this paper is to analyze a gap in the evaluation of classifiers and identify the cause of mislabeled data points. Our contributions include:

1) Identifying a measure of disagreements across classifiers, which can be used to quantify the performance of an ensemble of classifiers.
2) Applying the measure of disagreements to three packet classifiers trained on public traffic traces and demonstrating the accuracy of the ensemble increases by 14.63% when discarding packets with disagreeing predictions.

The remainder of the paper is organized as follows. §II presents prior research followed by §III that covers reliable inference using our measure of disagreement. Our evaluation results and disagreement insights are discussed in §IV. The paper is concluded in §V.

## II. Background

Obtaining visibility into the identity of connected devices is a network management task that can be done actively or passively. Active techniques send specific packets to target devices that are classified based on their responses. However, this technique is based on the assumption that the device sends truthful responses. Compromised, infected, or rogue

devices, however, can deceive monitoring tools by spoofing their responses. Passive techniques overcome this issue by using the traffic to/from the device as input data for their classification. However, shortcomings include (i) limited traffic to facilitate fast identification or (ii) spoofed traffic. Given their resilience, recent research has increasingly focused on passive techniques utilizing some of the knowledge from the general traffic classification [7].

Passive techniques match features or raw time-series signals from device traffic against known signatures or patterns. There are four broad types of methods used for these techniques: (a) Port-based [8], (b) Payload-based [9], (c) Statistical-based [10], and (d) Behavioral-based [11]. Port-based classification is the simplest technique. It maps port numbers to applications and classifies a device based on those applications. Payload-based techniques inspect the contents of packets and match the payload patterns to signatures learned from training devices. Statistical-based techniques analyze the raw network traffic to construct input features for data mining algorithms, such as a measure of average, min, or max packet size and/or count. In the data mining process, patterns in the data are found and matched against patterns that have been extracted from the training data. Behavioral-based methods identify a device by monitoring all the network traffic that the device sends/receives to build a behavioral profile. These profiles are then compared with each other and the device is labeled based on profile similarity [12].

In practice, the ground truth labels are not always available. This makes it difficult to determine whether the predictions are accurate and reliable. Previous research assumes that uncertain predictions must be rejected as they are most likely incorrect [13]. To purify predictions, [13] applies thresholds to the confidence score of models. [5] aggregates the output labels by MAC address and then replaces the labels with the mode. We argue that the level of variance among the results returned by different classifiers can be used to decide whether to accept or reject the device type predictions when monitoring a network.

A gap in the current literature is the lack of studies on when and why mispredictions occur, and a lack of evidence to prove that uncertain predictions are incorrect. Also, the reported performance metrics (*e.g.,* accuracy, F1 score) provide limited insights into the causes of classifiers predicting traffic classes. We: (i) propose a disagreement metric that can be applied to predictions from several models to discover the root cause of the misclassification, (ii) analyze whether leveraging predictions from other models reveals mispredictions or confirm less-confident predictions, and (iii) discuss possible methods of using disagreements as an out-of-distribution detection technique.

## III. Reliable Inference with Multiple Classifiers

To investigate the reliability of models, we experiment with three IoT packet classifiers from previous research [4]–[6]. These classifiers employ a diverse set of machine learning algorithms trained on features from packet headers.

TABLE I
Distribution of network packets across IoT device classes in the UNSW dataset.

| IoT class | # packets |
|---|---|
| Dropcam | 2.1m |
| Samsung SmartCam | 966k |
| Belkin Wemo Motion Sensor | 749k |
| Amazon Echo | 705k |
| Belkin Wemo Switch | 612k |
| Insteon Camera | 500k |
| Netatmo Welcome | 369k |
| Withings Smart Baby Monitor | 350k |
| Smart Things | 290k |
| Withings Aura Smart Sleep Sensor | 239k |
| TP-Link Day Night Cloud Camera | 198k |
| HP Printer | 166k |
| Netatmo Weather Station | 130k |
| Triby Speaker | 111k |
| LiFX Smart Bulb | 88k |
| Nest Dropcam | 76k |
| PIX-STAR Photo-Frame | 42k |
| iHome | 35k |
| TP-Link Smart Plug | 25k |
| Withings Smart Scale | 2985 |
| NEST Protect Smoke Alarm | 2317 |
| Blipcare Blood Pressure Meter | 131 |

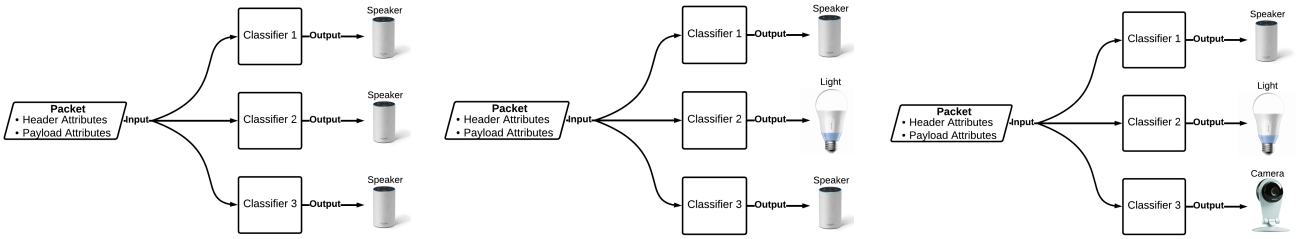
### A. Dataset

For this paper, we use the public packet traces of IoT devices from "UNSW dataset" [14], which has been widely used, curated, and validated in a variety of use-cases and IoT-focused research problems [5], [7], [11], [12], [14]. This dataset contains network traffic (in the form of PCAP files) of IoT and non-IoT devices over the course of two weeks collected from a lab testbed. We exclude non-IoT device traffic from our study as it is beyond the scope of this paper. Table I lists 22 unique IoT devices and their corresponding contribution (number of packets) to the UNSW dataset, consisting of a total of 7.8 m packets.

We note that this dataset contains both autonomously-generated and user-generated traffic. Also, there is no malicious or spoofed traffic, meaning noise is kept to a minimal level during the learning process for the classifiers.

We do not apply any class sampling techniques since [15] showed that such techniques do very little to improve the overall performance of the model. To ensure a comprehensive evaluation, we use a combination of metrics in our experiments (Accuracy, F1, Precision, Recall, Matthews coefficient). We did not make changes to the dataset after filtering out the packets that came from non-IoT devices. This was done by comparing the packet's source IP/MAC address against a list of known IoT IP/MAC addresses. We keep all IoT devices in our analysis, even those classes that are relatively less active such as the NEST Protect Smoke Alarm. The reason being classifiers are expected to handle such realistic heterogeneous classes and behaviors.

We split the dataset into 70% for training and the remaining 30% for testing. Note that the split is done temporally and not randomly. This approach was chosen to mimic real-




(a) No Disagreement. (b) Partial Disagreement. (c) Full Disagreement.

Fig. 1. Measure of disagreement: (a) no, (b) partial, and (c) full, across three IoT device classifiers.

world scenarios where data streams continuously. Doing so, we found that the training portion of this split does not contain any packets of Nest Dropcam; hence, this class is never seen/learned by our models. We will use this device (§IV-C) to see how the models predict an unseen class. In addition, three devices do not appear in the test data (Blipcare Blood Pressure Meter, PIX-STAR Photo-Frame and TP-Link Day Night Cloud Camera). These devices continue to serve a purpose in the experiment by increasing the difficulty for classifiers to accurately identify the correct class.

### *B. Chosen Classifiers*

We selected three models for our research based on the criteria of accepting packets as input, ease of implementation, ability to handle encrypted traffic, and not being trained on the UNSW dataset. In what follows, we provide a summary of each selected model.

**IoTSentinel:** This classifier [6] uses features from packet headers. [6] identifies IoT devices based on initial packets sent during a network onboarding process, such as associating with a WiFi gateway. A fingerprint matrix is created using 23 features from the first 12 packets in sequence. The majority of features are binary values indicating the use of certain protocols across the datalink, network, transport, and application layers. To avoid retraining the entire model, the authors trained a specific model per each known IoT class instead of a multi-class classifier.

**IoTSense:** This classifier [4] employed features from packet headers and payload. For packet header features, it uses the majority of binary protocol features in IoTSentinel. In total, 20 features were selected. IoTSense creates a 100-attribute fingerprint by extracting data from the first five packets of each session. This fingerprint was found to be best modeled by Gradient Boosting.

**IoTDevID:** Work in [5] started with a total of 112 packet features. The authors applied a filtering process to select the top 30 features. These features were then used to evaluate six independent classifiers to determine which was best suited for a real-time detection system. From the analysis, it was revealed that the decision tree had the highest accuracy and F1 score of 94.3% and 93.7%, respectively.

### *C. Classifier Adaptation*

Analysis of the dataset revealed two features used by IoTSense [4] and IoTSentinel [6], namely "`IP_padding`" and "`IP_alert`" had a value of zero throughout the entire dataset. These features relate to the Router Alert option in the IP header. Since they provided no information for the models, we excluded them. We also modified IoTDevID and IoTSense. Firstly, to diversify the modeling techniques used by our classifiers, we replaced the decision tree learner (DT) with a K-Nearest Neighbour (KNN) classifier for IoTDevID [5]. For IoTSense, we replaced the Gradient Boosting classifier with a Histogram-based Gradient Boosting classifier due to our large dataset (§III-A). This substitution had no impact on the accuracy but decreased the execution time.

For our experiments, we created the classifiers (Random Forest, KNN, and Histogram-based Gradient Boosting) that underlie the models using the Python package sklearn [16]. We configure hyperparameters with the values defined in their original experiments. The primary objective of this experiment is to combine multiple classifiers at different performance levels rather than focusing on optimizing their individual accuracy. We also note that customizing hyper-parameters for every data stream is practically infeasible in real-world settings, and models are often expected to be generalizable. The features identified by IoTSentinel [6], IoTSense [4], and IoTDev [5] are extracted from packet traces in the dataset [14] and recorded into three CSV files using the IoTDevID extraction code [17]. The CSVs are presented as input to their respective classifiers, where we quantify their performance and measure disagreement. The causes of the different levels of disagreement are also analyzed in the next section.

### *D. Measure of Disagreement Among Classifiers*

In this research, we present the "Disagreement" metric that represents the number of different labels assigned to a single instance. We define three different ways classifiers can disagree:

- **No Disagreement**: As shown in Fig. 1(a), all classifiers predict the same label for the input instance of traffic. This can also be interpreted as a full agreement.
- **Partial Disagreement**: As shown in Fig. 1(b), the majority of classifiers predict the same label.
- **Full Disagreement**: As shown in Fig. 1(c), none of classifiers share their predicted label.

A classified instance can only be in one of the three states. When more than three models are available, Partial Disagreement could be further broken up into "Majority Disagreement" when there is a plurality (less than 50% of classifiers agree),

TABLE II
PERFORMANCE OF INDIVIDUAL CLASSIFIERS WHEN APPLIED TO THE UNSW DATASET (TESTING PORTION: 30%).

| Classifier | **IoTSentinel** | **IoTSense** | **IoTDevID** |
|---|---|---|---|
| Accuracy | 73% | 80% | 81% |
| F1 | 76% | 81% | 82% |
| Precision | 84% | 86% | 85% |
| Recall | 73% | 80% | 81% |
| Matthews coefficient | 70% | 77% | 79% |

and "Minority Disagreement" when the majority of classifiers agree. Since this paper experiments with only three classifiers, the current definition of Partial Disagreement is sufficient.

In our experiments (§IV), the most common label will be selected as the final label when there is Partial Disagreement. The most confident prediction will be chosen when there is a Full Disagreement. The final labels may or may not align with the ground truth.

## IV. MODELS EVALUATION AND DISCUSSION

In this section, we analyze the measure of disagreement across the three models. We begin by quantifying the performance of the models by applying them to the UNSW dataset and highlighting differences in our results with those reported by the original works [4]–[6]. We also analyze how disagreements vary across IoT classes and communication protocols. We draw some insights into why the three classifiers disagree in their predictions. We then present how the measure of disagreement can be used to identify new classes of devices.

### A. Individual Classifier Results

Table II presents the performance metrics, including the overall accuracy, F1 score, precision, recall, and Matthews Coefficient of the individual classifiers. The values have been transformed into a percentage, and percentages are used throughout this paper.

In the original experiments, the accuracies for IoTSentinel, IoTSense and IoTDevID's decision tree were 81.5% [6], 99% [4] and 94.3% [5], respectively. Their performance, however, dropped 8-20% in our evaluations. Such performance deviations could be due to multiple reasons. Firstly, we apply no output purification. In the original experiments, all packets belonging to a MAC address were considered for classification at an aggregate level. This means that a prediction (*e.g.,* majority label) is assigned to a group of packets associated with a MAC address. For example, [5] used the mode of predictions as the final label for those packets. Instead, we measure the performance on a per packet basis. Additionally, the original models were trained and tested on a different dataset [18]. The UNSW dataset contains more varied traffic than the set-up traffic only Aalto dataset [18] and model performance can vary across different datasets [19].

### B. Ensemble Disagreements

This analysis aims to draw insights into the cause of various models predicting the class of packets differently. Overall, the classifiers together correctly labeled 82.15% of the packets, scoring higher than the individual models. We found that: more than 70% of tested packets (1.65m) received the same prediction from the three models (No Disagreement), where 97% of those were correct; 564K packets received two different labels (Partial Disagreement) – some are attributed to the relatively poor performance of IoTSentinel compared to the other two classifiers; and, 123K packets caused Full Disagreement. If we discard the packets with any disagreement, the accuracy goes up to 96.78% for individual packet labeling. Additionally, if packets with No Disagreement are aggregated and the mode label selected, the device classification accuracy is 94.74% (100% if out-of-distribution devices are not included). However, we are interested in the reason for disagreement, so we studied the quantity and quality of disagreements across devices and measured the disagreements across protocols. This is reported by Table III, Table IV, and Table V.

*1) Disagreement across Classes:* The Withings Smart scale experiences the highest level of disagreement, with 18.43% of its packets receiving three different labels from the classifiers, followed by the LiFX Smart Bulb at 13.72%. The Nest Dropcam was not considered because its packets were not in the training dataset. All other IoT classes see Full Disagreements in less than 10% of their packets. Withings Smart Scale had a relatively high level of disagreement because there were only 917 test packets. 169 of them caused Full Disagreement as there was a minimal amount of Smart Scale training data. Despite this, 157 of the Full Disagreement packets were correctly labeled. For the LiFX Smart Bulb, manual verification revealed the cause of the disagreements is due to the similarity to Smart Things. 72.78% of the Full Disagreement packets were DNS packets, and both LiFX Smart Bulb and Smart Things have identical patterns for their DNS packets. Since there are more DNS packets associated with Smart Things, a classifier predicted Smart Things over LiFX Smart Bulb with a confidence of 100%. However, another classifier correctly predicted Smart Things for these packets at a lower confidence (68%). The remaining 28.22% were TCP packets that caused disagreement, yet the most confident classifier correctly labeled them as LiFX Smart Bulb. In this study, we treat the models equally. However, to maximize accuracy, certain models could be weighted more on certain classes when their historical behavior has shown they consistently make correct predictions.

Dropcam had the least number of disagreements, with 97.51% of its packets receiving a shared prediction from the three models and 99.88% of those predictions were correct. This is most likely due to Dropcam's patterns being unique compared to the rest of the devices and having the greatest number of training packets. Such uniqueness leads to distinct traffic patterns, which the classifiers can easily distinguish. Generally, the closer a device's function is to that of another device, the more disagreements there are.

*2) Disagreement across Protocols:* Table V reports the measure of disagreement across communication protocols at the application, transport, network, and datalink layers. Among application layer protocols, Simple Service Discovery Protocol (SSDP), Dynamic Host Configuration Protocol (DHCP), and

TABLE III
OVERALL MEASURE OF DISAGREEMENT ON TESTING DATA ACROSS IOT CLASSES.

| IoT class | No Disagreement | Partial Disagreement | Full Disagreement |
|---|---|---|---|
| Dropcam | 97.51% | 2.48% | 0.01% |
| Samsung SmartCam | 68.77% | 27.32% | 3.90% |
| Belkin Wemo Motion Sensor | 64.58% | 32.37% | 3.05% |
| Amazon Echo | 77.24% | 22.14% | 0.62% |
| Belkin Wemo Switch | 63.22% | 34.54% | 2.24% |
| Insteon Camera | 60.57% | 35.60% | 3.83% |
| Netatmo Welcome | 41.75% | 57.28% | 0.97% |
| Withings Smart Baby Monitor | 73.12% | 25.83% | 1.04% |
| Smart Things | 81.20% | 17.82% | 0.98% |
| Withings Aura Smart Sleep Sensor | 31.33% | 61.60% | 7.07% |
| HP Printer | 23.77% | 66.53% | 9.70% |
| Netatmo Weather Station | 75.37% | 24.06% | 0.56% |
| Triby Speaker | 24.18% | 74.42% | 1.40% |
| LiFX Smart Bulb | 55.22% | 31.06% | 13.72% |
| Nest Dropcam **(Not in training set)** | 0.17% | 6.20% | 93.63% |
| iHome | 38.98% | 53.81% | 7.22% |
| TP-Link Smart Plug | 43.99% | 55.28% | 0.73% |
| Withings Smart Scale | 9.60% | 71.97% | 18.43% |
| NEST Protect Smoke Alarm | 63.86% | 34.30% | 1.84% |
| **Total** | 70.64% | 24.10% | 5.26% |

TABLE IV
MEASURE OF DISAGREEMENT ON TESTING DATA WHEN PREDICTION IS "CORRECT".

| IoT class | No Disagreement | Partial Disagreement | Full Disagreement |
|---|---|---|---|
| DropCam | 99.88% | 9.41% | 28.57% |
| Samsung SmartCam | 96.25% | 72.86% | 36.22% |
| Belkin Wemo Motion Sensor | 84.68% | 50.62% | 43.85% |
| Amazon Echo | 99.45% | 42.61% | 94.37% |
| Belkin Wemo switch | 88.64% | 58.41% | 0.85% |
| Insteon Camera | 98.39% | 69.63% | 99.63% |
| Netatmo Welcome | 98.33% | 26.55% | 8.92% |
| Withings Smart Baby Monitor | 100% | 96.53% | 100% |
| Smart Things | 100% | 0.05% | 100% |
| Withings Aura Smart Sleep Sensor | 99.76% | 62.71% | 24.87% |
| HP Printer | 99.99% | 79.77% | 14.90% |
| Netatmo Weather Station | 100% | 13.92% | 92.98% |
| Triby Speaker | 65.36% | 58.35% | 11.56% |
| LiFX Smart Bulb | 99.77% | 0.10% | 7.22% |
| Nest Dropcam **(Not in training set)** | 0% | 0% | 0% |
| iHome | 85.86% | 54.39% | 67.27% |
| TP-Link Smart Plug | 74.20% | 15.75% | 42.86% |
| Withings Smart Scale | 53.41% | 84.09% | 92.90% |
| NEST Protect Smoke Alarm | 84.17% | 78.57% | 58.33% |
| **Total** | 96.78% | 53.49% | 17.03% |

Hypertext Transfer Protocol Secure (HTTPS) packets receive more agreement (about 90% or more) from the three models. This means those three protocols display distinct patterns in their packets that the three models reliably learn. We found that the majority (about 93%) of SSDP packets in the UNSW dataset belong to the Samsung SmartCam, and 87% of the DHCP protocol came from the Netatmo weather station. Thus leading to $99.72\%$ of the SSDP packets and $92.76\%$ of the DHCP packets causing No Disagreement. HTTPS is used by only 11 of the devices and 75% of the packets are sent by Dropcam. This imbalance makes it easier to distinguish Dropcam packets. On the other hand, Domain Network System (DNS) packets received the largest Full Disagreement ($12.51\%$). This can be attributed to the fact that the features are dissociated to the payloads, and the headers of DNS packets often lack strong identifiable patterns. Finally, Network Time Protocol (NTP) packets have the most partial agreements as they also have no unique patterns in the dataset. $67.52\%$ were labeled correctly because the Insteon Camera made up $67.39\%$ of the testing data NTP packets.

In the transport layer, despite the lack of information extracted from User Datagram Protocol (UDP) packets, $87.45\%$ received full agreement (No Disagreement). This percentage is $10\%$ higher than for Transmission Control Protocol (TCP) packets. Full Disagreements for TCP packets ($6.66\%$) is more than five times ($1.27\%$) UDP packets. Further analysis revealed that most TCP packets with Full Disagreement contain zero payload, as they are parts of the TCP handshake. For example, 40-byte TCP packets are simple acknowledgments and resets. TCP packets with non-zero payloads comprise a small fraction of the full-disagreement cohort.

No strong insight can be drawn from the network-layer protocols. More than 45% of Internet Control Message Protocol (ICMP) packets get full agreement from the three models, while more than half lead to Partial Disagreements. The three models partially disagree in predicting ICMPv6 packets as HP Printer and Netatmo Welcome generate over 50% of the packets together. More than a third of ICMPv6 packets were labeled correctly, despite receiving low confidence scores of less than 40%. In this paper, we utilize raw confidence scores reported by individual models. These scores can be normalized to better interpret corresponding predictions; however, this is outside the scope of this study.

Finally, in the datalink layer, Logical Link Control (LLC) packets of IoT devices contain distinct patterns that are learned by our classifiers (All the packets had No Disagreement). More than 90% of LLC packets in our dataset are generated from the Netatmo weather station; thus, this situation is comparable to the SSDP packets mentioned above. Despite Address Resolution Protocol (ARP) packets having a greater balance across classes, the packets using this protocol received Partial Disagreement. More than 13% of the training ARP packets came from the Withings Smart Baby Monitor and since no distinct patterns were extracted by the feature set, most classifiers predicted ARP packets as Withings Smart Baby Monitor. Extensible Authentication Protocol over LAN (EAPOL) packets mainly originate from Netatmo weather station (40% of all EAPOL packets). Similar to ARP, EAPOL packets also do not have many unique statistical patterns for the classifiers to learn.

The main reason certain protocols have close to 100% No Disagreement or Partial Disagreement is that the feature sets do not yield unique patterns for those protocols. As there is no patterns to learn, the classifiers will predict the most common label for that protocol. IoTSentinel [6], IoTSense [4], and IoTDev [5] focus on extracting information from TCP and UDP protocols because of their predominant presence. Packet aggregation techniques are then used to correctly classify other protocols.

### C. Out-of-Distribution Detection

The Nest Dropcam packets were absent during training. Therefore, it became an out-of-distribution class for the classifiers during the testing phase and resulted in the greatest disagreement across the classes with $93.63\%$ of its packets causing Full Disagreement. Less than $0.2\%$ of the packets received No Disagreement. However, the packets that had full agreement were incorrectly labeled as Dropcam because they

TABLE V
MEASURE OF DISAGREEMENT ACROSS COMMUNICATION PROTOCOLS.

| Protocol | No Disagreement | Partial Disagreement | Full Disagreement |
|---|---|---|---|
| **Application layer** | | | |
| DNS | 70.77% | 16.72% | 12.51% |
| HTTPS | 88.77% | 3.02% | 8.21% |
| DHCP | 92.76% | 7.24% | 0% |
| MDNS | 77.81% | 18.44% | 3.75% |
| SSDP | 99.72% | 0.25% | 0.03% |
| NTP | 0.01% | 99.99% | 0% |
| **Transport layer** | | | |
| TCP | 77.35% | 16.59% | 6.06% |
| UDP | 87.45% | 11.28% | 1.27% |
| **Network layer** | | | |
| ICMP6 | 0.26% | 99.74% | 0% |
| IP | 78.78% | 15.37% | 5.85% |
| ICMP | 45.16% | 50.25% | 4.59% |
| **Data-Link layer** | | | |
| ARP | 0% | 100% | 0% |
| LLC | 100% | 0% | 0% |
| EAPOL | 71.44% | 20.35% | 8.21% |

were using HTTPS. Thus, the classifiers had a confidence between $50\%$ and $100\%$ for the No Disagreement labels. This shows that the confidence of the prediction does not necessarily correlate with its reliability.

This example shows that using the heuristic of the number of Full Disagreements being greater than the number of Partial and No Disagreements (*i.e.,* Full Disagreements $> 50\%$) could be useful in detecting out-of-distribution classes. Furthermore, this heuristic could be extended to monitor device behavior by using a sliding window. When the number of full disagreements in the window reaches $50\%$, then a new behavior has likely started. There will be a tradeoff with the window size as smaller sizes would detect new behavior quickly but be more susceptible to false positives. Larger windows would have the opposite attributes. Future work will verify the efficacy of this heuristic.

## V. CONCLUSION

The inference reliability of an ensemble can be improved by investigating the structure of disagreement. This paper has introduced a novel metric for measuring disagreement among packet classifiers and has demonstrated its utility for interpreting and rectifying predictions. By employing three models from previous research and applying them to a public dataset, we have uncovered insightful results regarding the agreement and disagreement among the models. Notably, approximately $70\%$ of packets in our dataset exhibited full agreement among the predicted labels from the three models, with an accuracy rate of over $96\%$. This finding indicates that when the models converge on a unanimous prediction, the reliability of the ensemble increases significantly.

Conversely, around $5\%$ of the packets resulted in full disagreement among the models, highlighting instances where the ensemble's reliability may be compromised. Intriguingly, more than $80\%$ of these packets were misclassified by the models. This indicates that the best strategy for device classification is to discard those predictions and rely only on No Disagreement predictions. Additionally, our analysis revealed that certain protocols, such as HTTPS, DHCP, and LLC, exhibited relatively lower levels of disagreement. These protocols displayed unique patterns in the packets of IoT devices, suggesting the presence of distinct characteristics that contribute to higher prediction agreement. However, these patterns will be impacted by the feature selection. By recognizing and leveraging these patterns with appropriate features, we can further improve the inference reliability of ensemble learners when classifying packets from these specific protocols.

## ACKNOWLEDGMENT

This research is supported by the Commonwealth of Australia as represented by the Defence Science and Technology Group of the Department of Defence.

## REFERENCES

[1] R. Al-amri *et al.*, "A Review of Machine Learning and Deep Learning Techniques for Anomaly Detection in IoT Data," *Applied Sciences*, vol. 11, no. 12, p. 5320, Jun 2021.

[2] G. Cirillo and R. Passerone, "Packet Length Spectral Analysis for IoT Flow Classification using Ensemble Learning," *IEEE Access*, vol. 8, pp. 138 616–138 641, Jul 2020.

[3] Y.-H. Hung, "Improved Ensemble-Learning Algorithm for Predictive Maintenance in the Manufacturing Process," *Applied Sciences*, vol. 11, no. 15, p. 6832, Jul 2021.

[4] B. Bezawada *et al.*, "Behavioral Fingerprinting of IoT Devices," in *Proc. ACM ASHES*, Toronto, Canada, Oct 2018, pp. 41–50.

[5] K. Kostas *et al.*, "IoTDevID: A Behavior-Based Device Identification Method for the IoT," *IEEE Internet of Things Journal*, vol. 9, no. 23, pp. 23 741–23 749, 2022.

[6] M. Miettinen *et al.*, "IoT SENTINEL: Automated Device-Type Identification for Security Enforcement in IoT," in *Proc. IEEE ICDS*, Atlanta, GA, USA, Jul 2017.

[7] H. Tahaei *et al.*, "The Rise of Traffic Classification in IoT Networks: A Survey," *Journal of Network and Computer Applications*, vol. 154, p. 102538, Mar 2020.

[8] G. Parra *et al.*, "Implementation of Deep Packet Inspection in Smart Grids and Industrial Internet of Things: Challenges and Opportunities," *Journal of Network and Computer Applications*, vol. 135, pp. 32–46, Jun 2019.

[9] M. Lotfollahi *et al.*, "Deep Packet: A Novel Approach for Encrypted Traffic Classification using Deep Learning," *Soft Computing*, vol. 24, no. 3, pp. 1999–2012, Feb 2020.

[10] O. Salman *et al.*, "A Machine Learning Based Framework for IoT Device Identification and Abnormal Traffic Detection," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 3, p. e3743, Sep 2022.

[11] A. Sivanathan *et al.*, "Detecting Behavioral Change of IoT devices using Clustering-Based Network Traffic Modeling," *IEEE IoT Journal*, vol. 7, no. 8, pp. 7295–7309, Mar 2020.

[12] A. Hamza *et al.*, "Verifying and Monitoring IoTs Network Behavior Using MUD Profiles," *IEEE TDSC*, vol. 19, no. 1, pp. 1–18, May 2022.

[13] A. Pashamokhtari *et al.*, "Inferring Connected IoT Devices from IPFIX Records in Residential ISP Networks," in *Proc. IEEE LCN*, Edmonton, AB, Canada, Sep 2021.

[14] A. Sivanathan *et al.*, "Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics," *IEEE Transactions on Mobile Computing*, vol. 18, no. 8, Aug 2019.

[15] S. Wang and X. Yao, "Multiclass Imbalance Problems: Analysis and Potential Solutions," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 4, pp. 1119–1130, Mar 2012.

[16] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, Oct 2011.

[17] K. Kostas *et al.*, "GitHub - kahramankostas/IoTDevID: A Behaviour-Based Fingerprinting Method for Device Identification in the IoT," 2022. [Online]. Available: https://github.com/kahramankostas/IoTDevID

[18] S. Marchal, "Iot devices captures," 2017. [Online]. Available: https://research.aalto.fi/en/datasets/iot-devices-captures

[19] M. Khan *et al.*, "Study and Observation of the Variation of Accuracies of KNN, SVM, LMNN, ENN Algorithms on Eleven Different Datasets from UCI Machine Learning Repository," in *Proc. iCEEiCT*, Dhaka, Bangladesh, Sep 2018.