CommuterScanner: Towards Smart Indoor Positioning Systems in Urban Transportation

Abhishek Mukherji†, Vinay Raghuram†, Santosh Pandey†, Huy Tran*†, Nirupama Bulusu*
†Cisco Systems, *Portland State University
{abhishmu, vraghura, sanpande}@cisco.com, {hptran, nbulusu}@pdx.edu

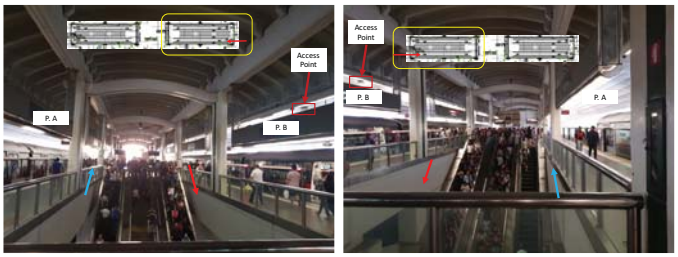
***Abstract*—With the rapid growth of modern cities, public transportation systems require smart planning to provide effective and competitive services to the daily commuters. Due to the presence of Wireless LAN Network (WLAN), Wi-Fi access is available to commuters during their daily commute. The overarching goal of this work is to leverage infrastructure-based indoor positioning systems (IIPS) deployed at train stations to enable several smart transportation use cases by analyzing commuter traffic. Specifically, in this work we address identification of whether a user is in-train or on-platform by utilizing two types of passively sensed Wi-Fi data, namely, received signal strength (RSSI) and phase vectors (AoA) measured at the deployed access points from data received from mobile devices. We conduct structured analysis of each data source, to identify features that distinguish on-platform and in-train devices. Our *CommuterScanner* solution achieves up to 90% accuracy using random forest model. Our solution works for a variety of deployments including APs with RSSI-only or RSSI+AoA capabilities and irrespective of if the device is connected to the Wi-Fi.**

I. Introduction

A. Indoor Positioning in Urban Transportation

With the rapid growth of modern cities, public transportation systems require smart planning to provide effective and competitive services to the daily commuters [18]. Due to the presence of Wireless LAN Network (WLAN) [6], Wi-Fi access is available to commuters during their daily commute [15], [25]. Infrastructure-based indoor positioning systems (IPS) utilize the WLAN and can identify devices on the stations by using received signal strength (RSSI) and angle-of-arrival (AoA) data measured for each device at the Wi-Fi access points. The key theme of this work is to leverage indoor positioning systems (IPS) deployed at the train stations to detect headcounts of commuters waiting for trains on different stations at any given time. This information may be used by transportation agencies to understand commuter traffic and enable smart transportation services.

Several smart transportation use cases can be enabled via IPSs deployed at stations such as (a.) adaptive scheduling of subways and commuter rails based on traffic, (b.) generating analytics such as determining traffic in each commute direction or weekday vs. weekend traffic, (c.) efficient and cost-effective air conditioning systems on stations and trains based on the headcounts, and (d.) emergency preparedness and evacuation planning. Overall goal is two-fold, namely, (a.) to provide effective and safe experience for daily commuters, and (b.) potentially save a city millions of dollars per year.



(a) Station view 1 (b) Station view 2

Fig. 1: IIPS deployment at a station.

B. Challenges of Commuter Classification in IPS

Transportation agencies are interested in analytics based on the accurate head counts of commuters waiting for train on each station. Further understanding the head count of commuters on train in each direction would be insightful as well. Overall, it is challenging to leverage existing IPS to accurately place the tracked devices on platform and train, respectively, due to two factors, namely, (a.) configuration of platforms and trains, and (b.) accuracy of IPS is limited.

Platform and train configurations. As shown in Figures 1 and 2, station platforms (marked in green in Figure 2) are often *narrow* and *long*. In our scenarios, the two stations were: S1= 14 meter wide $\times$ 148 meters long and S2= 23 meters wide $\times$ 68 meters long. The train on the tracks (marked in blue in Figure 2) is even narrower (approximately 3 meters wide, slightly shorter than the platform length). The platform also may have access to stair, escalators and elevators. Yet another factor, illustrated in Figure 3.a, is that majority of the commuters wait closer to the edge of their respective destination trains, rather than towards the center of the platform. The station depicted in Figure 1 has the escalators in the center, therefore, the area towards the center of the platform is even limited. In our tested stations, access points (APs) are only placed on the platform; no hotspots are available on trains. Other configurations are also possible, and would require configuration specific solutions.

Accuracy of IPS. As depicted in Figure 2, the RSSI-based indoor positioning systems have location estimates (depicted as red triangles) which can be up to 10 meters off from the actual device location [3], [6] (shown with dashed red circles). Similarly, AoA-based systems have an accuracy of 3 to 4 meters [21]–[23] (shown with dashed orange circles). Due to such limitations of indoor positioning systems and the narrow

978-1-5386-7902-9/19/$31.00 ©2019 IEEE

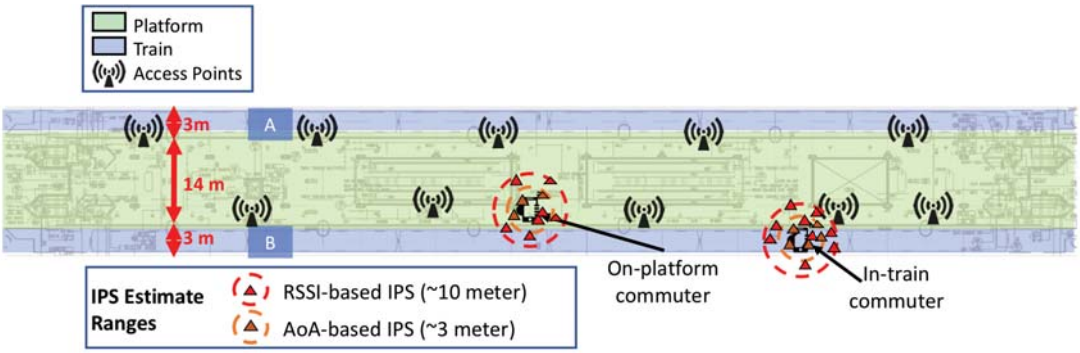


Fig. 2: CommuterScanner: on-platform vs. in-train classification problem.

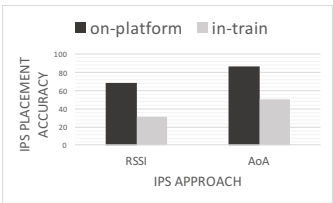
structures of platform and trains, the indoor positioning systems, by themselves, can only distinguish on-platform versus in-train devices with low accuracy. As shown in Figure 3.b, based on the systematic data collected with ground truths from live IPS deployments, the percentage of correctly placed on-platform devices by RSSI-based IPS is only 68%, whereas the same for AoA based IPS is 86%. However, due to the narrower trains, the percentages of correctly placed in-train commuters are 31% for RSSI-based IPS and only 50% for AoA-based IPS. *Therefore, in this work we focus on the problem of on-platform versus in-train classification by using the passively sensed RSSI and AoA (phase vectors) data received by IPS from devices via the Wi-Fi access points.*

Prior works [12], [24] have proposed using mobile device sensors such as accelerometer and GPS for transportation mode detection. However, use of device-based approaches would require particular apps or services installed on mobile devices of all commuters. Commuters may not be willing to install such apps due to numerous reasons such as battery drain from GPS use, privacy, etc. The app must work on variety of devices such as different iPhones, Android (Samsung, LG, OnePlus, etc.) and Windows phones, which typically vary also by the different sensors inside the devices. Often some of the sensors, such as GPS, cellular signal, may not work well when such commuter stations are underground. Therefore, we propose to leverage already deployed WLAN and infrastructure-based indoor positioning systems (IPS) by passively sensing signals from the mobile devices of the commuters.

C. Contributions

We present *CommuterScanner* system to classify if a user (device) is on-platform or in-train by using only RSSIs and phase vectors measured at the APs. Below, we describe the contributions of this work.

(a) Commuters waiting



(b) Correct placements by IPS

Fig. 3: Factors impacting commuter classification.

- To the best of our knowledge, our work is the first to examine RSSI and phase vectors for classification of device location in a commuter station. We leverage not only the instantaneous RSSI values from APs but also the phase vectors measured on the antenna array on each AP and across multiple APs for the given classification task.

- We conduct structured analysis of each data source, namely, RSSIs and phase vectors to identify features that distinguish on-platform and in-train devices.

- Our *CommuterScanner* solution achieves up to 90% accuracy using random forest and combining RSSI+AoA features. For RSSI only features up to 88% accuracy is achieved. Therefore, our solution works for RSSI only APs and APs with AoA capabilities. Thus, can be employed in a wide variety of deployments. Further, the solution works for both connected (using data packet RSSI and/or AoA) and unconnected devices (using probe-only RSSI).

II. Background

We first describe how infrastructure-based indoor positioning systems work, followed by the problem statement.

A. Infrastructure-based Indoor Positioning Systems

In this work, we use commercial-off-the-shelf (COTS) APs [6] that have been deployed at many enterprise setups such as airport, offices, malls, and hospitals. Figure 4 shows that the AP has 4 serving antennas (for transmitting or receiving signals from Wi-Fi devices) and 32 circular-array antennas. To localize a device, the infrastructure-based indoor positioning system (IIPS) uses two types of Wi-Fi measurements: received signal strength indicators (RSSIs) measured at the serving antennas and a phase vector (consisting of phase values) measured at the circular-array antennas. Each set of phase values is computed by using channel state information (CSI) [1] measured at the physical layer of the AP [11].

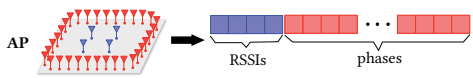


Fig. 4: Measurements at an AP [6]

Figure 5 illustrates an IIPS on top of an enterprise WLAN infrastructure [4]. Each AP measures Wi-Fi signals emitted

from tracked devices and forwards the measurements to a WLAN controller [5]. The controller aggregates and then forwards the measurements received from multiple APs to a location server (LS) deployed on-site or on-cloud. Based on the measurements, the LS uses a combination of RSSI-based trilateration and phase-based angle-of-arrival (AoA) method to localize and track the devices [11]. The method achieves median localization accuracy ranging from 1 m to 3 m for real-world deployments at many enterprise setups in retail, airports, and workplaces. AP density required is about one AP per 15 m x 15 m to guarantee good Wi-Fi coverage [10]. The method also achieves sub-meter accuracy for areas having line-of-sight between the device and several APs.

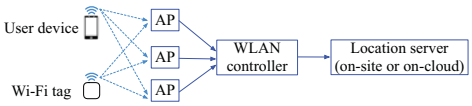


Fig. 5: Illustration of IIPS [4] on top of a WLAN

The IIPS samples Wi-Fi measurements without requiring an app installed on a mobile device. APs can measure RSSI for probing-only devices (not connected to Wi-Fi) [26]. For connected devices APs can measure both RSSI and phase values from data packets. For sampling RSSI measurements, an AP in the vicinity of a device measures the RSSI of the signal emitted from the device when the device sends a probe request to scan for APs [26] or a data packet to associated AP, or responds to a request from the AP if the device's radio transmitter is active [6]. Phase-based AoA localization requires a group of APs to measure the phases of the signal emitted from a device at the same time. Each group has a master AP which sends several request packets to all devices connected to itself in 250 ms [7]. The device sends response packets, which all APs in the group use to measure the phase. To localize all devices connected to different APs, each AP becomes master in a round-robin manner.

B. Problem Statement

Table I describes the notation used for defining the *CommuterScanner* on-platform vs. in-train detection problem.

Symbol	Description
$time_t$	Timestamp of time step t
est_t	Location estimate at time step t $\text{est}_t = [estX_t, estY_t]$
act_t	Actual location at time step t $\text{act}_t = [actX_t, actY_t]$
$channel_t$	Channel that a device is connected at time step t
RSSIs_t^i	= $[\text{RSSIA}_t^i, \text{RSSIB}_t^i]$, Signal strength in bands 5GHz (A) and 2.4GHz (B) measured at AP i at time step t
phases_t^i	Phase vector measured at AP i having antennas $1 \ldots P$ $[phase_t^{i,1}, \ldots, phase_t^{i,P}]$
a_t	$=[a_t^1, \ldots, a_t^i, \ldots, a_t^N]$, where $a_t^i = [\text{RSSIs}_t^i, \text{phases}_t^i]$, Data measured at N APs deployed on a floor at time step t
s_t	Device status at time step t $s_t = 1$: on-platform OR $s_t = 0$: in-train

TABLE I: Notation used for *CommuterScanner*.

On-platform vs. in-train detection. Figure 2 depicts the problem of detecting if user is on-platform or in-train. Given at current time t, $a_t^1, \ldots, a_t^n$ is the data (RSSIs and phase vectors) measured at multiple access points deployed on a station,

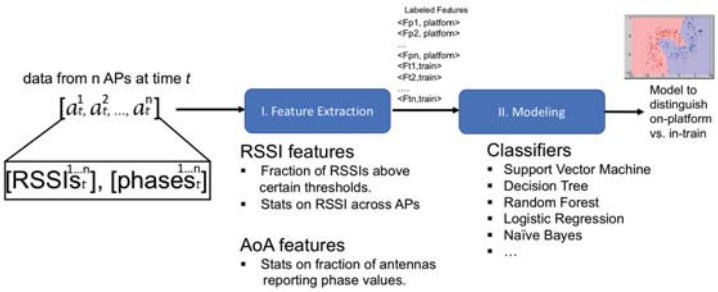


Fig. 6: *CommuterScanner* Architecture.

CommuterScanner predicts the device status s_t at t in real-time. In this paper, we interchangeably use words detection, prediction or classification to mean the same thing.

III. The *CommuterScanner* Approach

Here, we describe the overall *CommuterScanner* approach for detecting if a device is on-platform or in-train.

The key idea is to utilize RSSI and AoA (phase vectors) data received at any given time t, to identify if the device is on-platform or in-train. *CommuterScanner* is a machine learning framework designed as a two step process as illustrated in Figure 6. The first step is the feature extraction where all relevant features are extracted that potentially distinguish if a device is on-platform vs. in-train. The features are then tested with various machine learning models to determine which model will be best suited for the problem. Below we describe the two steps of (i.) feature extraction, and (ii.) modeling. As all APs are located on platform in our tested stations, we hypothesize that the APs would receive unobstructed and much stronger signals from on-platform devices than the in-train devices due to the metal body of the train.

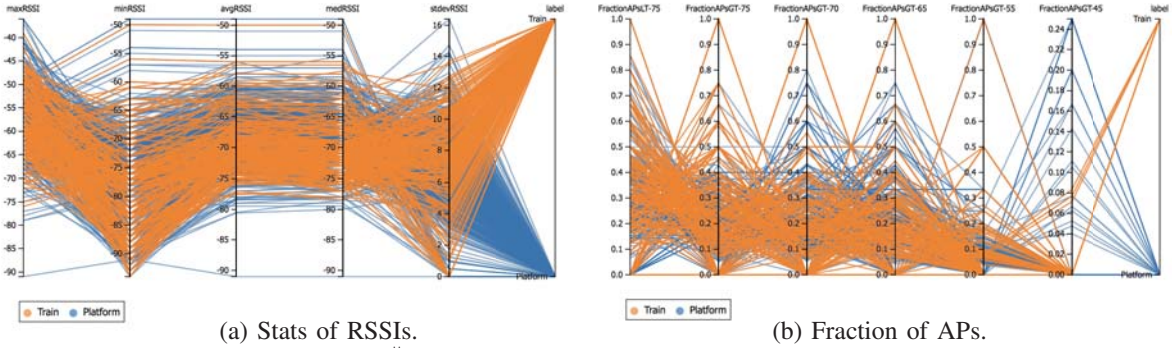
A. CommuterScanner Architecture

The model is trained using data collected on-platform and in-train. The model is then deployed on the IIPS for real-time predictions. The predictions are made for a device at each instance of input received from the APs. and, in case, the consecutive predictions jump between on-platform and in-train, a windowed majority voting based smoothing can then be applied on the prediction output to mitigate the jumpiness.

B. Feature Extraction

For a device, we extract and analyze the following features on RSSIs and phase vectors received from multiple APs.

RSSI features. The APs deployed on the station receive RSSI data from each device periodically. We explored two types of RSSI based features. First set of features are statistics over RSSIs across the reporting APs, such as maximum, minimum, average, median and standard deviation of RSSIs. Further, based on our hypothesis of weaker signal strengths for in-train devices compared with on-platform devices, we generated features such as the fraction of reporting APs that have



(a) Stats of RSSIs. (b) Fraction of APs.

Fig. 7: || co-ordinates view of RSSI-based features.

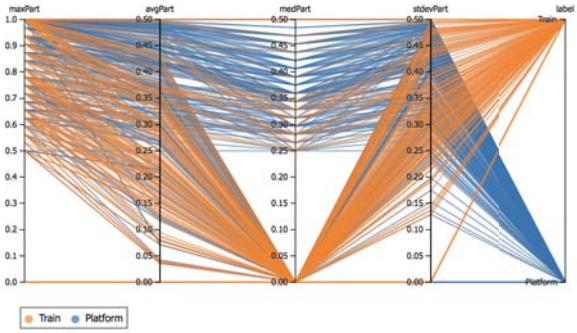


Fig. 8: || co-ordinates view of AoA-based features.

RSSI in certain range. RSSI ranges considered are {[-120,-75],(-75,-70],(-70,-65],(-65,-55],(-55,-45],(-45,0]}[1]. In Figures 7 (a) and (b), we depict the full set of RSSI-based features for in-train and on-platform devices. We utilize parallel co-ordinates [9] visualization to view how the features distinguish the labels. Each vertical line depicts an attribute (RSSI-based features and label) with different values marked on the line (e.g., *maxRSSI* varies from -35 to -90). Each horizontal line going across the attributes up to the label (train and platform) depicts a tuple. Data collected in-train is *orange*, and on-platform data is in *blue*. Most of the features have significant overlap between on-platform and in-train devices. However, maximum RSSI across APs is one feature where on-platform devices can sometimes be heard with up to -40 or stronger RSSI, while maximum RSSI for in-train devices mostly remains weak. Further, fraction of APs reporting RSSI in range (-45,0] has marked difference between in-train vs. on-platform devices. These observations conform to our hypothesis about the difference in signal strength, however, as shown in Figure 7, there is no single feature to distinguish between on-platform vs. in-train devices.

AoA features. Similar to the RSSI data, in AoA-based location computation, each AP computes a phase vector of n antennas (typically 16 or 32 based on AP model) for each client. Often, the phase vector may be partial, i.e., the vector may not have data from some of the antennas. For AoA data, we extend our hypothesis that the phase vectors may be received from fewer antennas inside the train than those on the platform. Thus, for each AP, we compute the fraction of antennas (out of 32) for which valid phase value is received. Statistics such as maximum, minimum, average, median, and standard deviation over the phase fractions across APs are computed (Figure 8). As depicted by the parallel co-ordinates view, no single attribute clearly distinguishes the in-train (orange) versus on-platform (blue) features.

[1]In notation (a,c], (a,c] is used to indicate an interval from a to c that is *exclusive* of *a*, but *inclusive* of *c*.

C. Modeling

Once the features are extracted, this step scales these features to the range $[0, 1]$ before applying a model to classify motion. We considered different models: (a.) Support Vector Machine (SVM) [8], (b.) Decision Tree (DT) [19], and (c.) Random Forest (RF) [14]. We also tried Logistic Regression and Naive Bayes but exclude them due to poor accuracy.

IV. Evaluation

Below, we describe our evaluation of *CommuterScanner* on-platform versus in-train detection approach. We collected data from infrastructure-based indoor positioning systems deployed at two stations.

A. Goals and Metrics

1) Goals: To investigate how our proposed *CommuterScanner* approach can be used for on-platform vs. in-train classification, we focus on answering the following questions:

Goal 1: Which model provides accurate classification using the different feature sets?

Goal 2: How effective are RSSI and AoA features in distinguishing on-platform vs. in-train devices?

2) Metrics: Table III below describes the metrics used in our evaluation. These metrics are well-defined in literature, yet we describe them in the context of our classification problem (see Table II).

Predicted/Actual	on-platform	in-train
on-platform	True Positive (TP)	False Positive (FP)
in-train	False Negative (FN)	True Negative (TN)

TABLE II: *CommuterScanner* confusion matrix.

Metric	Description
Training Accuracy	Average 10-fold cross validation accuracy for 80% training set.
Precision	$\text{Precision}_{\text{on-platform}}$ = TP / (TP+FP), $\text{Precision}_{\text{in-train}}$ = TN / (TN+FN).
Recall	$\text{Recall}_{\text{on-platform}}$ = TP / (TP+FN), $\text{Recall}_{\text{in-train}}$ = TN / (TN+FP).
f1 score	$\text{f1}_{\text{on-platform}} = 2*(\frac{(Precision_{\text{on-platform}}*Recall_{\text{on-platform}})}{(Precision_{\text{on-platform}}+Recall_{\text{on-platform}})})$ $\text{f1}_{\text{in-train}} = 2*(\frac{(Precision_{\text{in-train}}*Recall_{\text{in-train}})}{(Precision_{\text{in-train}}+Recall_{\text{in-train}})})$
Test Accuracy	Accuracy for the 20% test set.

TABLE III: Metrics for our evaluation based on Table II.

B. Data Collection

The goal of the data collection is to collect the measured (RSSIs and phase vectors) from the APs at the indoor positioning system from a connected mobile device together with the actual user label (on-platform or in-train). The indoor positioning system is instrumented to dump out the measured data systematically in a log file, which is then parsed and relevant information is extracted as features with label.

Stations	Station 1 14m×148m 10 APs at height=3.0m	Station 2 23m×68m 6 APs at height=3.5m)
Number of Participants	4	
Devices	iPhone6s, iPhone 7, Samsung S6, Nexus 6	
Minutes of data	platform= 245, train=196	platform= 210, train=158

TABLE IV: Data Collection.

We collected data at two commuter stations, we call them station 1 and station 2. The stations are similar in shape, with a platform in between two tracks for train A and train B going in opposite directions. Station 1 is a slightly longer setup with APs aligned on either sides of the platform for maximum Wi-Fi and location coverage. Table IV summarizes the dataset that we use for building our *CommuterScanner* prediction models.

Fig. 9: Avg. accuracy for 10-fold CV on 80% training set.

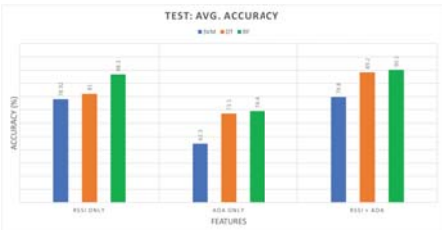


Fig. 10: Accuracy on 20% test set.

C. Experimental Results

We compare the accuracy of different models by using the data collected in 2 stations. Given the data collected in multiple experiments, we first randomly permute the experiments performed in each station. Then, we put 80% of the data into a training set for training the models and the remaining 20% of the data for testing the models. For Goal 1, we compare the three models, namely, (a.) SVM, (b.) Decision Tree, and (c.) Random Forest. To evaluate Goal 2, we test 3 combinations of features, namely, (a.) RSSI only, (b.) AoA only, and (c.) RSSI + AoA.

Figure 9 depicts the training accuracy. We perform a 10-fold cross validation on the training set. When using only RSSI features, SVM achieves 78.3% accuracy. Whereas decision tree and random forest achieve 81.8% and 86.5% accuracy, respectively. Overall, RSSI only features outperform AoA only features. The combination of RSSI + AoA feature slightly improves decision tree and random forest models. Trends are similar in the 20% test set (Fig. 10); Random forest model with RSSI+AoA achieves 90.1% accuracy on the test set.

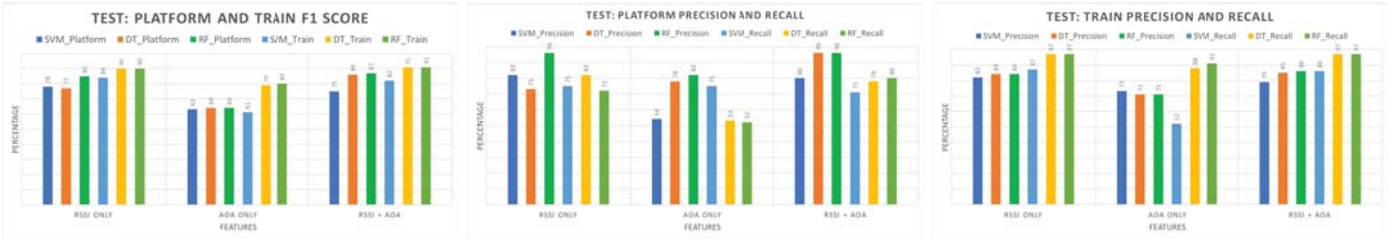
Figures 11 (a), (b) and (c), depict the f1 scores, precision and recall for on-platform and in-train, as defined in Table III. In particular, compared with the accuracy of indoor positioning (Fig. 3(b)), f1 scores for random forest model with RSSI only is 85% for on-platform and 90% for in-train. By combining RSSI+AoA based features, f1 score of random forest is 87% for on-platform and 91% for in-train. Therefore, *Commuter-Scanner* has greatly improved the classification accuracy, in particular, in detecting in-train devices.

Conclusions. CommuterScanner achieves up to 90% accuracy using random forest and combining RSSI+AoA features. For RSSI only features up to 88% accuracy is achieved. Therefore, our solution works for RSSI only APs and APs with AoA capabilities. Thus, can be employed in a wide variety of deployments. Further, the solution works for both connected (data packet RSSI and/or AoA) and unconnected devices (probe only RSSI).

V. Related Work

Transportation detection works are classified as (A.) client-based and (B.) infrastructure-based. *CommuterScanner* belongs to the second category.

A. Client-based transportation detection. Numerous efforts address the problem of detecting a transportation mode (walking, train, car, etc.) of a client. Client-side detection uses inertial sensor data (accelerometer, gyroscope, etc.) or wireless data (GPS, GSM, Wi-Fi) or the combination of both. Several works [12], [17], [27] extract features from inertial sensor data (accelerometer) to classify mobility (stationary or moving) and transportation mode of a client with accuracy above 90%. These approaches have high power consumption. Yu et al. [27] proposed a design to reduce the power consumption while achieving high accuracy. A client-side app is required and user consent to collect the sensor data. Wireless data based approaches [16], [24] extract features from wireless data such as Wi-Fi, GSM or GPS to classify transportation mode of a user with accuracy above 90%. GSM incurs smaller power consumption compared to GPS and is also applicable to indoor environments. Several works [2], [20] combine sensor and



20% Test Set: (a.) F1 Scores, (b) On-platform Precision and Recall, and (c) In-train Precision and Recall.

Fig. 11: Results for two stations.

wireless data. By using traditional models (decision tree, Hidden Markov model), Reddy et al. [20] achieve 93% accuracy. Chen et al. [2] present transfer learning based approach which is generalizable across different user devices.

B. Infrastructure-based transportation detection. These approaches perform the transportation detection by using the Wi-Fi signal heard opportunistically from a device. Kang et al. [13] proposed a system that uses a Wi-Fi monitoring device deployed on a bus to detect if a client gets on or off the bus. Our work focuses on data reported from access points deployed on a train station to detect if a user is on-platform vs. in-train. Infrastructure-side can be deployed while avoiding power consumption at client-side and no need for a mobile app.

VI. Limitations, Future Work and Conclusion

This work has the following limitations.

- How the classifier will be leveraged for headcount is omitted due to lack of space.
- Accuracy and generalizability of classifiers must be tested with more data from different station configurations.
- Additional factors such as time correlation may improve classification accuracy, and would be a great future study.
- Direction of train for in-train commuters can be detected.

In summary, we present *CommuterScanner* classification over RSSI and phase vector measurements for IIPS deployed in train stations. We demonstrate that our solution works for both RSSI-only and AoA based deployments; also irrespective of if device is connected to Wi-Fi.

References

[1] Y. Chapre, A. Ignjatovic, A. Seneviratne, and S. Jha. Csi-mimo: Indoor wi-fi fingerprinting system. In *39th Annual IEEE Conference on Local Computer Networks*, pages 202–209, 2014.

[2] Z. Chen, S. Wang, Z. Shen, Y. Chen, and Z. Zhao. Online sequential elm based transfer learning for transportation mode recognition. In *Cybernetics and Intelligent Systems (CIS), IEEE Conference on*, pages 78–83. IEEE, 2013.

[3] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan. Indoor localization without the pain. In *Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking*, MobiCom '10, pages 173–184, New York, NY, USA, 2010. ACM.

[4] Cisco. Cmx deployment models. https://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Borderless_Networks/Unified_Access/CMX/CMX_DeploymentModels.html, December 2015.

[5] Cisco. Cisco wireless controllers. https://goo.gl/TSgCLk, 2016.

[6] Cisco. Hyperlocation. https://goo.gl/B7RoRq, December 2017.

[7] Cisco. Cisco CMX Configuration Guide. https://goo.gl/NcZxGT, June 2018.

[8] C. Cortes and V. Vapnik. Support vector machine. *Machine learning*, 20(3):273–297, 1995.

[9] Y.-H. Fua, M. O. Ward, and E. A. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In *Proceedings of the Conference on Visualization '99: Celebrating Ten Years*, VIS '99, pages 43–50, 1999.

[10] Hyperlocation deployment guide by Cisco. https://goo.gl/x5swFg, December 2017.

[11] B. D. Hart, P. J. Stager, S. Pandey, D. Kloper, D. Lyons, and M. A. Silverman. Angle of arrival location sensing with antenna array, Nov. 21 2017. US Patent 9,823,330.

[12] S. Hemminki, P. Nurmi, and S. Tarkoma. Accelerometer-based transportation mode detection on smartphones. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, page 13. ACM, 2013.

[13] L. Kang, B. Qi, and S. Banerjee. A wireless-based approach for transit analytics. In *Proceedings of the 17th International Workshop on Mobile Computing Systems and Applications*, HotMobile '16, pages 75–80, 2016.

[14] A. Liaw, M. Wiener, et al. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

[15] Free wi-fi at all mrt lrt stations by 2020. https://www.gov.sg/news/content/today-online-free-wifi-at-all-mrt-lrt-stations-by-2020, Jan 2018.

[16] M. Mun, D. Estrin, J. Burke, and M. Hansen. Parsimonious mobility classification using gsm and wifi traces. In *Proceedings of the Fifth Workshop on Embedded Networked Sensors (HotEmNets)*, 2008.

[17] T. Nick, E. Coersmeier, J. Geldmacher, and J. Goetze. Classifying means of transportation using mobile sensor data. In *Neural Networks (IJCNN), The 2010 International Joint Conference on*, pages 1–6. IEEE, 2010.

[18] H. Poonawala, V. Kolar, S. Blandin, L. Wynter, and S. Sahu. Singapore in motion: Insights on public transport service level through farecard and mobile data analytics. In *ACM SIGKDD*, pages 589–598, 2016.

[19] J. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, Mar 1986.

[20] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava. Using mobile phones to determine transportation modes. *ACM Transactions on Sensor Networks (TOSN)*, 6(2):13, 2010.

[21] R. Roy and T. Kailath. Esprit-estimation of signal parameters via rotational invariance techniques. *IEEE Transactions on acoustics, speech, and signal processing*, 37(7):984–995, 1989.

[22] R. Schmidt. Multiple emitter location and signal parameter estimation. *IEEE transactions on antennas and propagation*, 34(3):276–280, 1986.

[23] Z. Shan and T.-S. Yum. A conjugate augmented approach to direction-of-arrival estimation. *IEEE Transactions on Signal Processing*, 53(11):4104–4109, 2005.

[24] L. Stenneth, O. Wolfson, P. S. Yu, and B. Xu. Transportation mode detection using mobile phones and gis information. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '11, 2011.

[25] Going online while underground: 12 cities compared. https://www.pcmag.com/article2/0,2817,2489108,00.asp, Jan 2018.

[26] E. Union Agency for Network and Information Security. Passive wifi surveillance and access point hijacking. https://www.enisa.europa.eu/publications/info-notes/passive-wifi-surveillance-and-access-point-hijacking, 2015.

[27] M.-C. Yu, T. Yu, S.-C. Wang, C.-J. Lin, and E. Y. Chang. Big data small footprint: the design of a low-power classifier for detecting transportation modes. *Proceedings of the VLDB Endowment*, 7(13):1429–1440, 2014.