Close-Range Indoor Proximity Detection for COVID-19 Exposure Notifications Using Smartphone Magnetometer Traces

Zach Van Hyfte and Avideh Zakhor
Dept. of Electrical Engineering and Computer Sciences
University of California, Berkeley
{zachvanhyfte, avz} @berkeley.edu

***Abstract*—Smartphone apps for exposure notification and contact tracing have been shown to be effective in controlling the COVID-19 pandemic. However, there is evidence that the Bluetooth Low Energy approach used for proximity detection by existing apps can be error-prone in areas with high numbers of metallic objects. In this paper, we present a new method for detecting whether or not two smartphones are 2 or fewer meters apart, intended to augment BLE-based proximity detection methods. We design a set of binary machine learning classifiers that take as input pairs of 10-second-long segments of magnetometer traces. These classifiers identify pairs of trace segments for which the two recording devices were 2 or fewer meters apart for at least 75% of the segment duration. We introduce a simple method of compensating for different magnetometer biases in heterogeneous devices. We confirm that our classifiers can generalize well to both new buildings and new devices whose traces are not present in their training data, and characterize their overall accuracy for heterogeneous-device proximity detection to be between 93.1% and 96.2%.**

***Index Terms*—proximity detection, contact tracing, magnetometer traces, machine learning, heterogeneous devices**

I. Introduction

The COVID-19 pandemic sparked a proliferation of "exposure notification" and "contact tracing" smartphone apps designed to alert users if they came within close proximity of an individual infected with COVID-19 [1], [2], [3], [4]. Most of these apps continuously track, in an oblique and privacy-preserving manner, which other devices a user's smartphone has been near. When a user tests positive for COVID-19, those other devices notify their users of their potential exposure to the virus. Authors in [1], analyzing the effects of the NHS COVID-19 app for England and Wales, conclude that the app was effective in reducing the spread of the virus.

Currently, the majority of exposure notification and contact tracing solutions are based on Bluetooth Low Energy (*BLE*) token broadcasts, although approaches that augment BLE with ultrasound broadcasts [5] and with analysis of ambient sound fingerprints [6] have also been proposed. The maximum distance from which a BLE token can be received varies, but can be significantly longer than the social distancing threshold of roughly 2 meters established by the U.S. Centers for Disease Control and Prevention (CDC) in 2020. Furthermore, evaluations of an app that uses the Google/Apple Exposure Notification system in areas with high numbers of metallic objects have demonstrated that the system does not properly identify nearby devices in those areas [7], [8].

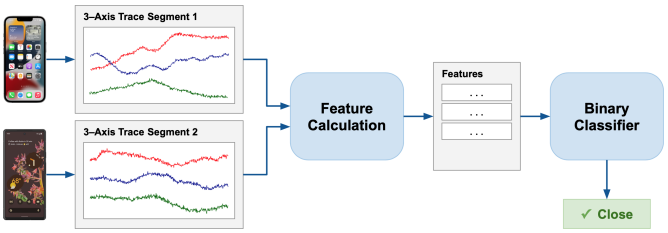

Fig. 1. The high-level design of our magnetometer trace–based proximity detection method.

In this work, we present a new magnetometer-based method for detecting whether or not two smartphones are in immediate physical proximity, i.e. 2 or fewer meters apart. Specifically, we design and evaluate a set of binary machine learning classifiers that take as input two 10-second segments of smartphone magnetometer traces. These classifiers identify pairs of trace segments for which the two recording devices were 2 or fewer meters apart for at least 75% of the segment duration. Our method is intended to enhance the accuracy of smartphone-based exposure notification and contact tracing apps; it is designed to be robust in dealing with inputs from heterogeneous devices and from a wide range of physical environments, with a low impact on device battery life. Figure 1 shows the high-level design of our magnetometer trace–based proximity detection method.

Disturbances in the Earth's magnetic field — caused by the steel used in the construction of many larger buildings, as well as by metallic objects and furniture — introduce distortions into smartphone magnetometer readings. The pattern of distortions seen in a device's magnetometer readings over time serves as a distinct "fingerprint" of the particular path being walked through a building. Over the past decade, a number of systems that localize devices within a building using sequences of magnetometer readings over time, i.e. magnetometer "traces," have been proposed and evaluated in the literature, with a typical average error of $0.5 - 3$ meters [9], [10], [11]. Because it requires devices to be close enough to buildings or metallic objects for the devices' magnetometer readings to be affected, this method of proximity detection is only usable indoors.

[12] presents a method of detecting whether two devices are in close physical proximity by comparing segments of their magnetometer traces, also intended for use in digital

contact tracing. However, to measure the similarity between two trace segments, this method relies only on the Pearson correlation coefficient of the norms of the three-element magnetic field vectors that make up the segments; a threshold value is empirically determined based on the length of the trace segments being compared, and a pair of segments is classified as "Close" if the Pearson correlation coefficient of their constituent magnetic field vectors' norms is above that threshold. In contrast, our methods make use of additional pre-processing steps, as well as a much wider variety of similarity measurements that account for similarities or dissimilarities between each of the three components of the magnetic field vectors, in addition to the norms; we use machine learning algorithms to, in effect, automatically determine the combination of features and threshold values that yields the highest proximity detection accuracy. Furthermore, the authors in [12] only evaluated their approach on pairs of segments that either (a) had the exact same trajectory, or (b) were collected in two distinct areas, often in two different buildings. We go beyond this, evaluating our methods on sets of segment pairings with a wider variety of distances between their two constituent segments, for which each pair of segments were recorded in the same building. As opposed to differentiating between segment pairings with the same exact trajectory and segment pairings recorded in entirely different areas, the classifiers that we present perform proximity detection at a finer level of granularity, with a more diverse set of "Close" traces; they differentiate between pairings recorded 2 or fewer meters apart on average and those recorded further apart but still within Bluetooth range of each other. Lastly, while all of the traces used in the evaluation of the method from [12] were collected with the same smartphone model, we introduce a way of compensating for different magnetometer biases in heterogeneous devices, and evaluate our methods with this added mitigation by training and evaluating classifiers on different subsets of traces from 4 different smartphone models.

The remainder of this paper is structured as follows: In Section II, we describe our approach and the composition of our training and evaluation sets; in Section III, we introduce a simple method of compensating for different magnetometer biases in heterogeneous devices and present experiment results; in Section IV, we conclude the paper and outline directions for future work.

II. Approach

In this section, we detail how the magnetometer trace data used in our experiments was collected, how groups of uniform-length segments are extracted from traces for use in training and evaluation sets, which types of segment pairs make up our training and evaluation sets, and lastly which features are calculated for each segment pair and passed as inputs to the classifiers.

Data Collection and Preparation: We used traces from the MagPIE dataset [13], [14], as well as traces we collected ourselves on the UC Berkeley campus, to create training and evaluation sets for our experiments.

The MagPIE dataset contains raw traces from a single smartphone's magnetometer, recorded in 3 different buildings across the campus of the University of Illinois Urbana-Champaign. We only used the traces that were recorded while a person walked holding the smartphone; we did not use the traces that were recorded while the smartphone was mounted on a wheeled robot. We also did not use any of the "live load" traces — i.e. the traces that were recorded with additional metallic objects placed along the path being walked to make the magnetic field in the area different than it was when other traces were recorded in the same area.

In order to be able to evaluate how well our methods perform on inputs from heterogeneous devices, we also collected new magnetometer trace data in Cory Hall, a five-story academic building on the UC Berkeley campus, using 4 different Android smartphones and the same tools employed in the creation of the MagPIE dataset [13], [14]. On two separate days, we collected data in a roughly 30m × 15m lab space on the third floor of the building. Magnetometer traces from a Samsung Galaxy S8, an Oppo RX17 Pro, a Google Pixel, and a Google Pixel 3 [1] were recorded using a slightly modified version of the "MagnetometerV2" data collection app [15] released by the authors of the MagPIE dataset [13], [14]. Ground-truth position data was simultaneously recorded using the MagPIE authors' "BRG_Trajectory" app [16] — which uses the Google Tango API to continuously localize a device within an indoor space — on a Tango-equipped Lenovo Phab 2 Pro.

Similar to the traces we used from the MagPIE dataset, the magnetometer traces we collected in Cory Hall were recorded while a single person walked a number of intersecting linear and non-linear paths while holding the Phab 2 Pro and one of the 4 test phones at a time. The person who recorded the Cory Hall traces navigated around hallways, cubicles, desks, office chairs, electronic and mechanical equipment, and computers; the smartphone recording the trace was always held flat in the palm of one hand in a fixed orientation, and the Phab 2 Pro was held vertically upright, with its rear cameras facing forward, in the other hand. The sets of paths walked with the individual devices were broadly similar, but not entirely the same; likewise, the set of paths walked on the first day of data collection was different from the set of paths walked on the second day.

In both the MagPIE and Cory Hall datasets, the magnetometer log file and corresponding position log file for a given trace do not share the same sequence of entry timestamps. Before transforming the traces into training and evaluation sets, for each trace, we match each magnetometer reading with a temporally "nearby" position measurement — i.e. a position measurement whose timestamp is within 0.01 seconds of the magnetometer reading's timestamp — from the corresponding position log file, using the procedure described in Section 3.3 of [17].

Segment Extraction: We extract segments of a uniform temporal length L_S from each magnetometer trace to use in training and evaluation sets. Two sets, or "tracks," of segments are extracted from each trace, where one track is offset from the other by $\frac{L_S}{2}$ seconds. Each track corresponds to a particular starting position in the trace; the starting position for the first track is 0 seconds and the starting position for the second track is $\frac{L_S}{2}$ seconds. For each track, the trace is divided into contiguous, non-overlapping segments of length L_S. For example, for a 25-second trace with $L_S = 10$

[1] Running Android versions 9, 8.1, 10, and 10, respectively

seconds, the segments from the first track would cover the intervals $[0\text{s}, 10\text{s}]$ and $(10\text{s}, 20\text{s}]$, while the segments from the second track would cover the intervals $[5\text{s}, 15\text{s}]$ and $(15\text{s}, 25\text{s}]$, as shown in Figure 2. Compared to a sliding-window method, this method of segment extraction yields fewer segments, and thus smaller training and evaluation sets. However, it bounds the amount of overlap between segments — a given pair of segments have at most 50% of their magnetometer readings in common, and a given pair of segments from the same track do not share any magnetometer readings. As detailed the next section, we only pair up segments from tracks with the same offset.

Segment Length: For all of the experiments detailed in Section III, we set the segment length $L_S = 10$ seconds. This time interval is long enough to allow distinct signatures of different paths to appear in segments, but short enough that if two users were close together long enough for transmission to occur, that period of contact would represent a significant portion of the segment. In addition, this segment length is within the range of time intervals that the authors in [12] determined yielded the highest overall performance when used with their method, as shown in Figure 9 of [12]. Using a short segment length also allows a single system to easily adapt to a wide range of contact duration thresholds for different infectious diseases or variants; to determine if two individuals were in immediate physical proximity for some arbitrary amount of time, a system can simply keep a running total of the number of L_S-second intervals for which the two were estimated to have been in immediate physical proximity.

Proximity Classes: To prepare training and evaluation sets from a given pool of segments, we first isolate segments into different groups. We assigned each segment from the MagPIE dataset to one of 3 groups based on the building in which it was recorded. Similarly, we assigned each segment from the Cory Hall dataset to one of 2 groups based on the day on which it was recorded.

We then enumerate every possible pairing of two distinct segments that are both from the same group and from tracks with the same offset — in a real-world proximity detection system based on our methods, only segments recorded at or around the same time would be input to the classifiers, and BLE can likely be used to filter out segment pairs recorded in entirely different buildings. For each segment pair (S_i, S_j), we iterate through every entry of S_j; for each entry of S_j, we find the temporally closest entry in S_i and calculate the two-dimensional distance d between the two entries' ground-truth positions. We then assign each segment pair a "proximity class" based on an analysis of all of the pair's entry distances d. If the percentage of entries for which $0 \text{ m} \leq d \leq 2.25 \text{ m}$ is at least P, the segment pair's proximity class is set to "Close." If the percentage of entries for which $3.25 \text{ m} \leq d \leq 20 \text{ m}$ is at least P, it is set to "Far." For both of the sets of experiments detailed in Section III, we evaluate our classifier design for both $P = 75\%$ and $P = 95\%$.

Segment pairings are dropped from the training and evaluation sets if neither of the aforementioned two criteria are met. This excludes from the training and evaluation sets any pairs of segments for which a significant portion of pairs of corresponding entries were recorded more than 20 m apart on average. We do this in an effort to focus the training process on the fine-grained differentiation between segment pairs recorded in immediate physical proximity and pairs recorded in somewhat close physical proximity, i.e. Bluetooth range. This is in contrast to an approach which would optimize the classifiers for the coarse-grained differentiation between pairs recorded in immediate physical proximity and pairs recorded very far away from each other. The fact that we only consider segment pairs (S_i, S_j) where S_i and S_j are from tracks with the same offset guarantees that none of the entries in S_i also appear in S_j, so that the segment pairs in our training and evaluation sets more closely mimic real-world inputs. Although our classifiers are only trained on segment pairs recorded $\leq$ 20 m apart, their accuracy in distinguishing "Close" segment pairs from segment pairs that were recorded $>$ 20 m apart matches or exceeds their accuracy in distinguishing "Close" pairs from pairs that were recorded between 3.25 m and 20 m apart, as shown in Section III.

Sensing Frequency: The sensing frequency of the traces from the MagPIE dataset is roughly 50 Hz. The Samsung Galaxy S8 and Google Pixel used to create the Cory Hall dataset also record traces at 50 Hz, while the Oppo RX17 Pro and Google Pixel 3 record traces at 100 Hz. Before calculating the features described below, we downsample all traces to 2 Hz. This is a practical sensing frequency for a real-world exposure notification app, with a low impact on device battery life — tests in [18] found that continuously sensing at 2 Hz reduced the battery life of a Samsung Galaxy S5 by only 4%, compared to 24% when sensing at 50 Hz.

Classifier Input Features: Once all of the segment pairs have been enumerated, and every pair has been assigned a proximity class, we calculate a set of features for each segment pair. These features are the inputs that are actually passed directly to the classifiers and used by them to predict the proximity classes of segment pairs.

Each magnetometer reading has three components: X, Y, and Z, which represent the magnetic field intensity measured by the magnetometer along the X, Y, and Z axes defined by Android's `Sensor` APIs [19]. For a given magnetometer trace segment S, let $|S|$ denote the number of entries in S. Define $M_{S,i}[\text{x}]$, $M_{S,i}[\text{y}]$, and $M_{S,i}[\text{z}]$ as the X, Y, and Z components of the magnetometer reading in the i-th entry of the segment S. Additionally, define $|M_{S,i}|$ as the norm of the magnetic field vector in the i-th entry of S.

Let $M_{\text{X}}(S)$ be a vector containing only the X-components of the magnetometer readings in the entries of S, in the same order as their corresponding entries:

$$M_{\text{X}}(S) \;=\; \Big[\; M_{S,1}[\text{x}],\; M_{S,2}[\text{x}],\; M_{S,3}[\text{x}],\;\; \ldots \;\; M_{S,|S|}[\text{x}] \;\Big]$$

Define $M_{\text{Y}}(S)$ and $M_{\text{Z}}(S)$ similarly. Likewise, let $M_{\text{N}}(S)$ be a vector containing only the norms of the magnetometer readings in the entries of S, in the same order as their corresponding entries.

Consider a pair of trace segments, (S_i, S_j). For each pair of single-axis vectors — $(M_{\text{X}}(S_i), M_{\text{X}}(S_j))$, $(M_{\text{Y}}(S_i), M_{\text{Y}}(S_j))$, and $(M_{\text{Z}}(S_i), M_{\text{Z}}(S_j))$ — and additionally the pair of norm vectors $(M_{\text{N}}(S_i), M_{\text{N}}(S_j))$, the cosine similarity as well as the Pearson, Spearman, and Kendall correlation coefficients of the two segments' vectors are passed as input features to the classifiers. These similarity measurements are

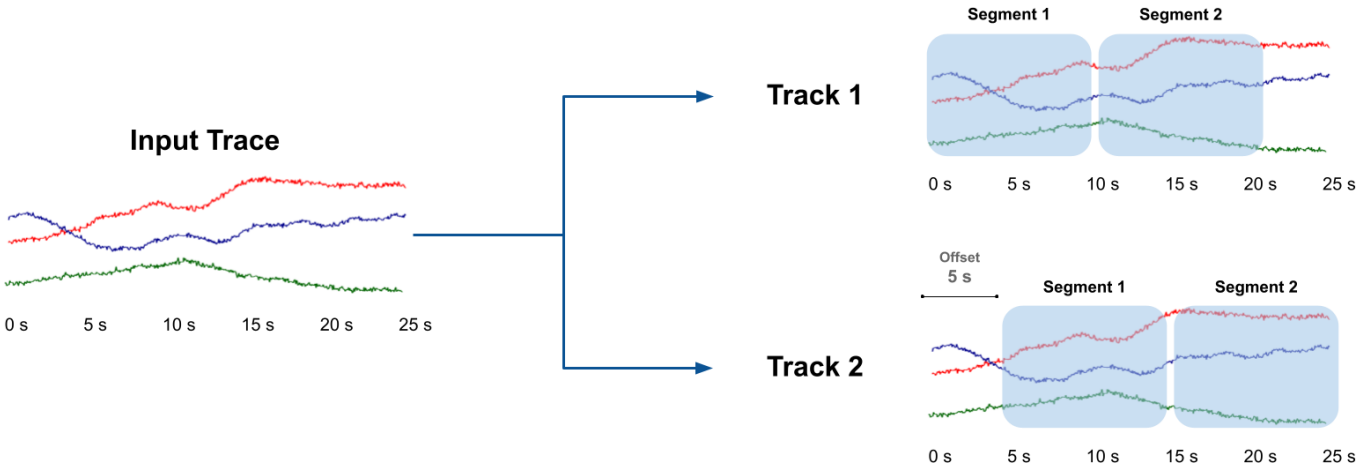

Fig. 2. An example of our segment extraction process, showing all 4 of the segments that would be extracted from a 25-second trace.

intended to provide the classifiers with indications of how well the fluctuations in the measured magnetic field strength in one segment match or align with the fluctuations in the other segment.

Furthermore, for a pair of segments (S_i, S_j), let $D_{\mathrm{X}}(S_i, S_j)$ be a vector containing the differences between the X-components of the magnetometer readings in the entries of S_i and S_j:

$$D_{\mathrm{X}}(S_i, S_j) = \left[M_{S_i,k}[\mathrm{x}] - M_{S_j,k}[\mathrm{x}] \ \ldots \ \forall\, k \leq \min(|S_i|, |S_j|) \right]$$

Define $D_{\mathrm{Y}}(S_i, S_j)$ and $D_{\mathrm{Z}}(S_i, S_j)$ similarly, and let $D_{\mathrm{N}}(S_i, S_j)$ likewise be a vector containing the differences between the norms of the magnetometer readings in the entries of S_i and S_j. For each of these individual vectors of differences — $D_{\mathrm{X}}(S_i, S_j)$, $D_{\mathrm{Y}}(S_i, S_j)$, $D_{\mathrm{Z}}(S_i, S_j)$, and $D_{\mathrm{N}}(S_i, S_j)$ — the mean, median, and standard deviation of the vector's elements are passed as input features to the classifiers. These features are intended to provide the classifiers with a summary of how far apart the magnetometer readings in the two segments tend to be along each axis, and an indication of whether or not this difference between their readings tends to remain the same over time — i.e. whether or not the fluctuations along a particular axis in each segment follow a similar pattern.

Segment Alignment: In our empirical analysis of pairs of segments from the two datasets, we found that, in a significant number of "Close" segment pairs, both segments contain a very similar pattern of magnetic field strength fluctuations, but the pattern begins at a different time within each segment, such that one segment "lags" another, usually by at most 1 second. This phenomenon is likely the result of two users walking alongside each other for some time, with one slightly ahead of the other.[2] To correct for this, as detailed in Section 3.5.1 of [17], we find the optimal value Δ^*, where -2 sec $\leq \Delta^* \leq 2$ sec, such that shifting S_i by Δ^* seconds relative to S_j results in the highest correlation of the norms of the magnetometer readings. Then, we calculate all of the features described above for the shifted segment pair; thus, the full input to the classifiers consists of two sets of the features described above — one for the original, unshifted S_i and S_j, and another for the shifted S_i and S_j.

[2]Or, in this specific case, considering how the datasets were collected, one person walking very close to where they had previously walked while recording a different trace.

III. Experiment Results

In this section, we first train and evaluate classifiers on sets of traces from the MagPIE dataset, which were recorded across 3 different buildings using a single smartphone. Then, we introduce a simple method of compensating for different magnetometer biases in heterogeneous devices, and evaluate our approach with this added mitigation by training and evaluating classifiers on different subsets of the traces that we recorded with 4 different smartphone models in Cory Hall.

Homogeneous-Device Experiments: To determine how well our approach performs in a baseline setting without the complications introduced by heterogeneous devices, and to determine how well it generalizes to different buildings, we trained a set of three classifiers on data from the MagPIE dataset. Each classifier was trained on data from 2 of the 3 buildings in which traces were recorded, and evaluated on data from the remaining building. The total numbers of "Close" / "Far" segment pairs generated from the CSL, Loomis, and Talbot building traces were roughly 1,600 / 44,000, 1,400 / 19,000, and 700 / 12,000, respectively, when $P = 75\%$, and 1,200 / 32,000, 1,100 / 9,200, and 400 / 6,800 when $P = 95\%$. For each of the three cases, we created an evenly class-balanced training set consisting of all "Close" segment pairings from the 2 training buildings and an equal number of randomly selected "Far" pairings from those 2 buildings. We then used these training sets to train random forest classifiers — specifically, instances of the `RandomForestClassifier` class from scikit-learn [20] version 1.1.2, with the `bootstrap` option disabled and the rest of the hyperparameters kept at their default values. Finally, we evaluated each classifier on the entire set of "Close" and "Far" segment pairings from its corresponding evaluation building. Table I shows the results of these evaluations for both $P = 75\%$ and $P = 95\%$. The "TN" ("True Negatives") columns show the classifiers' true negative rates, i.e. the percentages of "Far" samples correctly identified as such. Likewise, the "TP" ("True Positives") columns show the classifiers' true positive rates, i.e. the percentages of "Close" samples correctly identified as such. Additionally, the "TN 20+" column shows the percentage of segment pairs correctly

TABLE I
RESULTS OF HOMOGENEOUS-DEVICE EXPERIMENTS USING OUR METHODS

Evaluation Building	P = 75%			P = 95%	
	TN 20+	TN	TP	TN	TP
CSL	99.9%	99.9%	69.1%	99.9%	72.3%
Loomis	91.7%	88.5%	99.2%	88.1%	98.7%
Talbot	98.3%	93.5%	93.1%	91.7%	94.5%
Average	96.6%	94.0%	87.1%	93.2%	88.5%

TABLE II
RESULTS OF HOMOGENEOUS-DEVICE EXPERIMENTS USING METHOD FROM [12]

Evaluation Building	P = 75%	
	TN	TP
CSL	70.3%	66.3%
Loomis	67.4%	75.0%
Talbot	69.3%	62.8%
Average	69.0%	68.0%

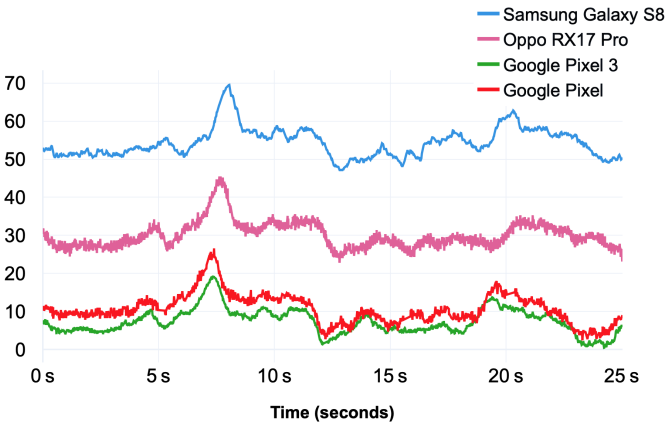

Fig. 3. The X components of the 4 test devices' magnetometer readings, in μT, over time as each of the devices was individually taken down the same corridor.

identified as such by each classifier when it was evaluated on the set of all segment pairs from its evaluation building for which at least 75% of pairs of corresponding entries were recorded at least 20 m apart. The classifiers perform well overall when evaluated on data from buildings other than those whose data was included in their respective training sets. Their performance also remains strong across both looser and stricter definitions of "Close" and "Far," i.e. lower and higher values of P, which indicates that our methods are adaptable to a range of different proximity definitions and contact duration thresholds.

As a basis for comparison, we also quantified the performance of the method presented in [12] for $P = 75\%$. As noted above, to determine the similarity between two segments, this method relies only on the Pearson correlation coefficient of the norms of the magnetic field vectors that make up the segments; a pair of segments is classified as "Close" if the correlation coefficient is higher than a threshold value θ_c. To perform this comparison, we trained another 3 `RandomForestClassifiers` like the ones described above, but with all of the input features removed except for the Pearson correlation coefficient of the norms of the segments' magnetometer readings — essentially, we used `RandomForestClassifiers` as a means of selecting the threshold values θ_c that yielded the highest overall performance. We then evaluated each of these new classifiers on the entire set of "Close" and "Far" segment pairings from its corresponding evaluation building, just as we did with the first set of classifiers. Table II shows the results of this evaluation; our methods substantially outperform the method presented in [12], with the true negative and true positive rates for our methods more than 19% higher on average than those for the method from [12].

Heterogeneous-Device Experiments: The magnetometers in different devices have different biases and sensitivities; examples of these can be seen in Figure 3, which shows the X components of our 4 test phones' magnetometer readings over time for the periods during which they were each individually taken down the same corridor in Cory Hall. The pattern of fluctuations is quite similar in all 4 lines, but there are fairly large vertical gaps between the lines, and there are small variations in the height of the "peaks" and depths of the "valleys" across the 4 traces. The Y components and Z components exhibit similar patterns.

To make our methods as robust as possible in the face of inputs from heterogeneous devices, we explicitly compensate for the biases of individual magnetometers by determining a single baseline magnetometer reading for each device, and then subtracting that baseline reading from all of the magnetometer readings in the traces recorded by that device.

To determine the baseline magnetometer readings for our 4 test phones, we used the MagnetometerV2 app to record the devices' magnetometer traces as they were all held steady in the exact same location — in the center of a field on the UC Berkeley campus, relatively far from buildings and other sources of magnetic anomalies — and in the same orientation in which they were held while traces were recorded in Cory Hall. We set the X, Y, and Z components of each phone's baseline reading to the average value of those respective components across all of the magnetometer readings in the phone's steady-position trace. This method of determining a baseline reading is fast and simple enough to be incorporated into the setup process for a real-world exposure notification or contact tracing app.

To evaluate how well our methods — including the baseline subtraction described above — perform in a heterogeneous-device setting, we trained a set of classifiers on the data that we recorded with our 4 test phones in Cory Hall. The total numbers of "Close" / "Far" segment pairs generated from the first and second day's traces from Cory Hall were roughly 590 / 4,200 and 450 / 2,200, respectively, when $P = 75\%$, and 370 / 2,200 and 200 / 900 when $P = 95\%$. For over 74% of these segment pairs, the two constituent segments were recorded by two different phone models. We trained 4 classifiers in total; each classifier was trained on segment pairs from 3 of the phones and evaluated only on segment pairs involving the remaining phone (the "evaluation device"). To compile the training and evaluation sets for each classifier, we first assigned all segment pairs for which at least one of the two segments was recorded by the evaluation device to the evaluation set. Then, from the remaining pairs — i.e. the pairs for which neither of the two segments was recorded by the evaluation device — we compiled an evenly class-balanced training set. This training set consisted of all of the

TABLE III
RESULTS OF HETEROGENEOUS-DEVICE EXPERIMENTS USING OUR METHODS

Evaluation Device	P = 75%			P = 95%	
	TN 20+	TN	TP	TN	TP
Galaxy S8	98.7%	97.8%	92.6%	97.8%	94.0%
RX17 Pro	98.4%	97.6%	91.9%	98.7%	93.0%
Pixel 3	98.6%	96.8%	95.6%	95.5%	96.6%
Pixel	98.9%	98.2%	87.9%	99.8%	88.3%
Average	98.7%	97.6%	92.0%	98.0%	93.0%

TABLE IV
RESULTS OF HETEROGENEOUS-DEVICE EXPERIMENTS USING METHOD FROM [12]

Evaluation Device	P = 75%	
	TN	TP
Galaxy S8	61.7%	59.5%
RX17 Pro	60.1%	67.2%
Pixel 3	58.1%	64.6%
Pixel	58.1%	62.5%
Average	59.5%	63.5%

"Close" segment pairs from the set of remaining pairings and an equal number of randomly selected "Far" pairs from that same set. Finally, we trained a `RandomForestClassifier` on this class-balanced training set, and evaluated it on the entire evaluation set. Table III shows the results of these heterogeneous-device experiments — the percentage of true negative samples correctly identified and the percentage of true positive samples correctly identified, for $P = 75\%$ and for $P = 95\%$. The results confirm that classifiers developed using our methods can distinguish between "Close" and "Far" segment pairs recorded by heterogeneous devices with high accuracy, and that their performance is solid even when evaluated on data from devices other than those whose data is included in the training set.

As with the homogeneous-device experiments, we compared the performance of our heterogeneous-device methods to that of the method presented in [12] for $P = 75\%$, training another set of 4 `RandomForestClassifiers` like the ones described above, with all of of the input features removed except for the Pearson correlation coefficient of the norms of the segments' magnetometer readings. We then evaluated each new Pearson coefficient–only classifier on the set of all segment pairs involving its corresponding evaluation device, just as we did with the first set of 4 classifiers. Table IV shows the results of this evaluation; the true negative rate and true positive rate for our methods are more than 37% and 28% higher on average than the ones for the method from [12], respectively.

IV. CONCLUSION AND FUTURE WORK

In this work, we presented a new magnetometer-based method for detecting whether or not two smartphones are 2 or fewer meters apart, intended for use in exposure notification and contact tracing apps. Our evaluations demonstrate that classifiers developed using our methods can distinguish between "Close" and "Far" segment pairs from non–tilt-compensated traces with over 90% balanced accuracy on average. Our results also show that our classifiers can generalize well to different buildings and devices whose traces were not present in their training data. Notably, the traces used in these experiments were not tilt-compensated. Tilt compensation would be necessary for a real-world proximity detection system, as the traces input to the system would be recorded in many different orientations. However, in our case, all 4 test devices were held in the same orientation during the recording of all of the traces, so tilt compensation is not strictly necessary for performing accurate comparisons of the traces. Future work will include using the accelerometer and gyroscope data recorded by the MagnetometerV2 app to tilt-compensate the traces, likely using one of several well-established tilt-compensation procedures.

REFERENCES

[1] C. Wymant, L. Ferretti, D. Tsallis, M. Charalambides, L. Abeler-Dörner, D. Bonsall, R. Hinch, M. Kendall, L. Milsom, M. Ayres, C. Holmes, M. Briers, and C. Fraser, "The epidemiological impact of the NHS COVID-19 App," *Nature*, 2021.

[2] Government of Singapore, "TraceTogether," 2020. URL: `https://www.tracetogether.gov.sg`.

[3] California Department of Technology, "CA Notify," 2020. URL: `https://canotify.ca.gov`.

[4] WeHealth, "Covid Watch Arizona," 2020. URL: `https://www.wehealth.org/arizona`.

[5] P.-S. Lo, "Accuracy of Bluetooth-Ultrasound Contact Tracing: Experimental Results from NOVID iOS Version 2.1 Using Five-Year-Old Phones," June 2020. URL: `https://www.novid.org/downloads/20200626-accuracy.pdf`.

[6] G. Bahle, V. Fortes Rey, S. Bian, H. Bello, and P. Lukowicz, "Using Privacy Respecting Sound Analysis to Improve Bluetooth Based Proximity Detection for COVID-19 Exposure Tracing and Social Distancing," *Sensors*, vol. 21, no. 16, 2021.

[7] D. J. Leith and S. Farrell, "Measurement-based evaluation of Google/Apple Exposure Notification API for proximity detection in a commuter bus," *PLOS ONE*, vol. 16, pp. 1–16, 04 2021.

[8] D. J. Leith and S. Farrell, "Measurement-based evaluation of Google/Apple Exposure Notification API for proximity detection in a light-rail tram," *PLOS ONE*, vol. 15, pp. 1–16, 09 2020.

[9] S. Shahidi and S. Valaee, "GIPSy: Geomagnetic indoor positioning system for smartphones," in *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–7, 2015.

[10] H. J. Bae and L. Choi, "Large-Scale Indoor Positioning using Geomagnetic Field with Deep Neural Networks," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pp. 1–6, 2019.

[11] S.-C. Yeh, W.-H. Hsu, W.-Y. Lin, and Y.-F. Wu, "Study on an Indoor Positioning System Using Earth's Magnetic Field," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 3, pp. 865–872, 2020.

[12] S. Jeong, S. Kuk, and H. Kim, "A Smartphone Magnetometer-Based Diagnostic Test for Automatic Contact Tracing in Infectious Disease Epidemics," *IEEE Access*, vol. 7, pp. 20734–20747, 2019.

[13] D. Hanley, A. B. Faustino, S. D. Zelman, D. A. Degenhardt, and T. Bretl, "MagPIE: A Dataset for Indoor Positioning with Magnetic Anomalies," in *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–8, 2017.

[14] B. R. Group, "Magnetic Positioning Indoor Estimation (MagPIE) Dataset." URL: `http://bretl.csl.illinois.edu/magpie`.

[15] "MagnetometerV2.tar.gz — Powered by Box." URL: `https://app.box.com/s/8g155irfv4ldbqcc5arnzhld3d9d62ig`.

[16] A. Faustino, "`alex-faustino/BRG_Trajectory`: Android app for a google tango enable device.," 2017. URL: `https://github.com/alex-faustino/BRG_Trajectory`.

[17] Z. Van Hyfte, "Proximity Detection Using Wi-Fi Fingerprints and Smartphone Magnetometers, With Applications to COVID-19 Surveillance," Master's thesis, EECS Department, University of California, Berkeley, May 2022.

[18] S. Kuk, J. Kim, Y. Park, and H. Kim, "Empirical Determination of Efficient Sensing Frequencies for Magnetometer-Based Continuous Human Contact Monitoring," *Sensors*, vol. 18, no. 5, 2018.

[19] "Sensors Overview — Android Developers." URL: `https://developer.android.com/guide/topics/sensors/sensors_overview#sensors-coords`.

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-Learn: Machine Learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011.