MAGPRINT: Deep Learning Based User Fingerprinting Using Electromagnetic Signals

Lanqing Yang [†#], Yi-Chao Chen [†#],Hao Pan[†], Dian Ding[†], Guangtao Xue[†*], Linghe Kong[†], Jiadi Yu[†], Minglu Li [‡†]
[†] Department of Computer Science and Engineering, Shanghai Jiao Tong University, China
[‡] College of Mathmatics and Computer Science, Zhejiang Normal University, China
Email:{yanglanqing,yichao,panh09,dingdian94,gt_xue,linghe.kong,jdyu,mlli}@sjtu.edu.cn

***Abstract*—Understanding the nature of user-device interactions (e.g., who is using the device and what he/she is doing with it) is critical to many applications including time management, user profiles, and privacy protection. However, in scenarios where mobile devices are shared among family members or multiple employees in a company, conventional account-based statistics are not meaningful. This poses an even bigger problem when dealing with sensitive data. Moreover, fingerprint readers and front-facing cameras were not designed to continuously identify users. In this study, we developed MAGPRINT, a novel approach to fingerprint users based on unique patterns in the electromagnetic (EM) signals associated with the specific use patterns of users. Initial experiments showed that time-varying EM patterns are unique to individual users. They are also temporally and spatially consistent, which makes them suitable for fingerprinting. MAGPRINT has a number of advantages over existing schemes: i) Non-intrusive fingerprinting, ii) implementation using a small and easy-to-deploy device, and iii) high accuracy thanks to the proposed classification algorithm. In experiments involving 30 users, MAGPRINT achieves 94.3% accuracy in classifying users from these traces, which represents an 10.9% improvement over the state-of-the-art classification method.**

***Index Terms*—electromagnetic induction; user fingerprinting; mobile device**

I. INTRODUCTION

Gartner [1] forecasted that there will be more than 9 billion cellphones, tablets, and laptops by the end of 2020, which is equivalent to 1.15 devices for every person on earth. Understanding the nature of user-device interactions (e.g., who is using the device and what he/she is doing with it) is critical to many analytic systems. For example, foreground Apps can be monitored to detect the current activity of the user. Likewise, the time spent on various Apps can be used to assess the efficiency of workers and systems [2] for use in time management [3]. It is even possible to infer the interests and personality of users by assessing the type of Apps they habitually use [4].

Researchers have demonstrated that user-device interactions can be used for user fingerprinting. The fact that users hold and use their mobile devices in distinct ways means that data from the built-in accelerometer can be used to infer the identity of the user [5]. Power consumption [6] and acoustic signals [7] have also been used as side-channels for user fingerprinting.

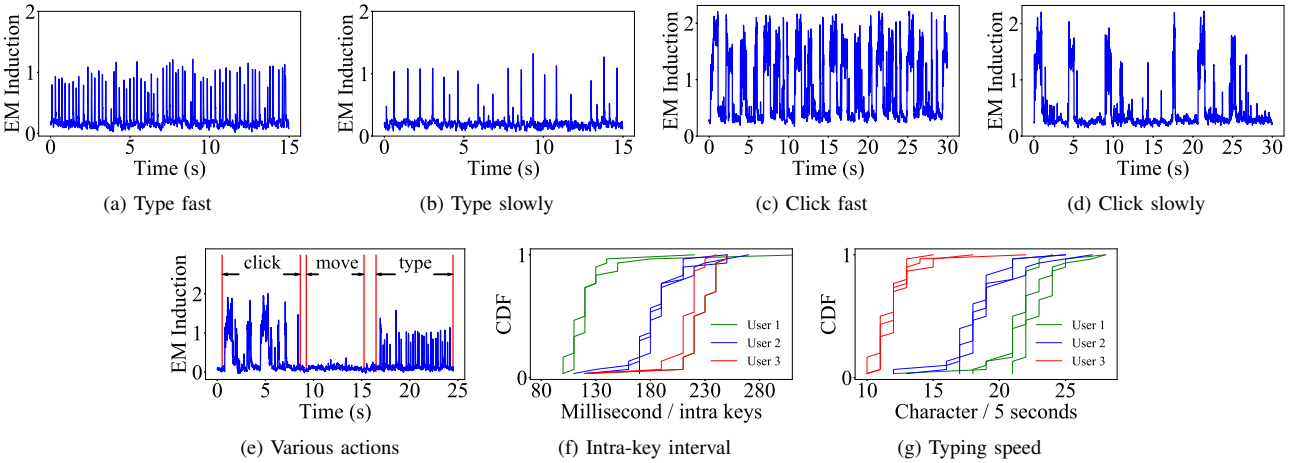
In many situations, it is critical to know the identity of the individual currently using the mobile device. For example, children have been known to pay for games using their parents' credit card [8], [9]. User-device interaction data could be used to ensure the individual attempting to use the credit card is indeed the owner. Likewise, sensitive data has been stolen when employees failed to log out of their user accounts [10], [11]. However, existing user authentication methods, like fingerprint readers and front-facing cameras, were not designed to continuously identify users. It would be highly beneficial to identify the current user in an on-going manner to ensure only the registered user has access to specific accounts or data.

In this study, we proposed to use electromagnetic (EM) signals as a side-channel for fingerprinting users. The intensity of EM signals generated from a mobile device reflects the computational intensity of the device. For example, a heavy computational load tends to push up power consumption by the CPU, which leads to an increase in EM induction. Furthermore, users vary considerably in the way that they use their devices in terms of key hold duration, typing interval, cursor speed, pause and click time, etc [5]–[7]. The instructions associated with these actions vary the internal operations of the device and are thus reflected in the EM signals.

In this paper, we developed the first system aimed at fingerprinting the EM signals emitted from mobile devices in order to infer the current user in real time. The proposed MAGPRINT system includes sensor hardware and an operational algorithm. The hardware is a small module comprising a magnetic sensor, computing unit, and wireless communication device. The module is easily attached to any mobile device for the collection of EM signals, regardless of the computing platform. MAGPRINT can also work with built-in magnetic sensors. The algorithm is designed to identify the user from the emitted EM signals due to the user-device behavior with a high degree of accuracy.

We encountered four fundamental challenges in designing MAGPRINT: 1) Raw EM signals include significant interference from Apps and tasks running in the background, which can greatly degrade the user identification accuracy. 2) Capturing some user characteristics tends to require a long-term observation; however, most user-device interactions are very brief. Thus, the algorithm must be able to capture EM signals of short interval as well as identify patterns that form

[#] Both authors contributed equally to the research.
[*] Corresponding author.



(a) Type fast (b) Type slowly (c) Click fast (d) Click slowly

(e) Various actions (f) Intra-key interval (g) Typing speed

Fig. 1: Electromagnetic (EM) signals generated when users used "Microsoft Word" on MacBookPro-2015. (a)-(e) show the EM patterns associated with various user-device interactions, including mouse clicking, mouse moving, and typing. (f) and (g) show CDF of the intra-key intervals and typing speeds of 3 volunteers in 3 different days.

over extended periods of time. 3) The EM signals associated with a given user tend to change over time and may even vary with their mood, thereby necessitating adaptive adjustment. 4) User characteristics also tend to vary with the specifics of the Apps. For example, the mouse is used more when browsing websites and typing operations are used more when sending a message. The enormous number of Apps that are in use makes it all but impossible to train an individual model for each of Apps, and pre-trained classification models function poorly when applied to unfamiliar Apps.

In this study, MAGPRINT designed algorithms to address these challenges. We first implement an interference cancellation method based on deep learning to mitigate the impact of the interference. Identification functions are handled by a novel network model, referred to as *TF-FCN-LSTM* (Time-Frequency Long Short Term Memory Fully Convolutional Network), for the extraction of short-term features as well as long-term use characteristics. To make the system robust, triplet loss function is adopted to extract features which are independent to time, users' mood, etc. Finally, Apps are classified according to usage characteristics in order to reduce the sample size required to train the model. Our objective was to ensure stable operations even in cases where the Apps are not included in the training set.

We implemented the prototype of MAGPRINT using a commodity magnetic sensor. To evaluate its accuracy, we conducted extensive experiments involving 30 volunteers using 30 the most popular Apps [12] on 10 mobile devices. MAGPRINT achieved an average of 94.3% in user identification based on EM fingerprints.

The main contributions of this paper are summarized in the following.

- This is the first report on fingerprinting users by tracking the magnetic signals emitted from mobile devices due to user-device interactions.
- We developed a deep learning-based interference cancellation method to filter out signals that are distinct from those of user interactions.
- We developed a *TF-FCN-LSTM* model with triplet loss function to enhance the reliability of fingerprinting, even when applied to unknown applications and users.
- Experiments on a prototype of the MAGPRINT system demonstrated that it significantly outperforms existing classification methods in terms of accuracy (by 10.9%) and robustness (Sec. VI-C).

The remainder of this paper is as follows. Sec. II presents a review of work in this field. Sec. III outlines our preliminary study illustrating how user habits result in unique EM patterns. In Sec. IV, we detail the design of MAGPRINT and present the *TF-FCN-LSTM* fingerprint model. In Sec. V, we detail the implementation of the MAGPRINT prototype. In Sec. VI, we assess the efficacy of the MAGPRINT system and present the experiment results. Conclusions and future work are summarized in Sec. VII.

II. RELATED WORK

The proposed scheme involves two operations: side-channel sensing and time-series classification for user fingerprinting.

Side-channel sensing: Side-channel sensing is widely used for the analysis of user-device interactions. Some previous studies employed the built-in sensors in smart phones to detect external hardware, such as keyboards [5], [7], computer hard drives [13], and 3D printers [14]. Coarse-grained monitoring of power consumption makes it possible to identify the Apps that are being used [6]. Yang *et al.* [15] demonstrated the use of data from USB charging stations to infer the history of websites that were visited. However, advancements in PCB design and packaging have made it far more difficult to monitor power consumption. In fact, monitoring schemes based on USB charging are rendered useless when the mobile

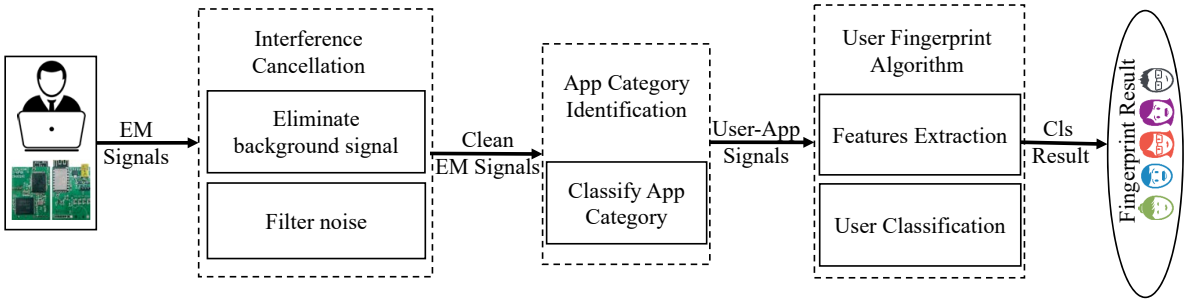


Fig. 2: MAGPRINT framework.

device is powered only by a battery. In [16] and [17], a near-field magnetic probe was used to extract the key of RSA applications operating on a Lenovo laptop; however, the size of such devices precludes their use with mobile devices. Wang *et al.* [18] employed a wrist-worn magnetic sensor and data acquisition device to recognize patterns in the usage of electrical appliances. MAGPRINT is able to obtain data of far finer resolution than the above-mentioned works based on magnetic sensors.

Deep-learning for time-series classification: Time-series classification (TSC) is an important and challenging problem in data mining. In the past, temporal features were extracted manually and then fed into a single classifier or multiple integrated classifiers to generate an output [19]–[22]. In recent years, there has been a shift toward deep neural networks to automate the extraction of features for time series classification tasks [23]–[28]. Cui *et al.* pioneered the multi-scale convolutional neural network (MCNN) [23]. Wang *et al.* used a fully convolutional network (FCN) for the classification of sequential data. Malhotra *et al.* [24] employed long short-term memory networks (LSTM) for anomaly detection from multiple sensors. The use of FCN for feature extraction in the FCN-LSTM model enabled Karim *et al.* [29], [30] to achieve state-of-art time-series classification performance. Compared to previous methods, the proposed MAGPRINT system is less sensitive to background noise. It also automates the generation of time-series features, while simultaneously enabling the extraction of features in the time-frequency domain, which makes it generalizable to a much wider range of devices and applications.

III. PRELIMINARY ANALYSIS

Preliminary experiments were conducted to answer three fundamental questions: i) Do user-device interactions generate detectable electromagnetic (EM) signals? ii) Do different users produce different EM patterns? iii) Are EM patterns spatially and temporally consistent for the same user? The answers to the above questions would clarify the feasibility of EM-based user fingerprinting.

Do user-device interactions generate detectable EM signals?

One volunteer was asked to conduct a variety of typing or mouse operations during which the operations and corresponding EM signals were recorded. As shown in Fig. 1e, each peak of EM signals during the "typing" period represents the typing of each character. For EM signals corresponds to "mouse moving", we can observe flat periodic signal. For those corresponds to "click mousing", we can see that each time we click the mouse, there are several irregular peaks. Therefore, the characteristics specific to each of these actions manifested as differences in the EM signals. Further experiments shown in Fig.1a-1d shows that, when moving or clicking the mouse and typing with different speed, the received EM signals can at the same time reflect these changes.

Do different users produce different EM patterns?

There are notable differences in the way that any given application (App) is used. Take Twitter as an example, some people compose many tweets, whereas others spend most of time reading the tweets composed by others or re-tweeting. Even when using the same App, individuals may vary in their user-device interactions, like keyboard intra-key timing, key hold duration, and typing interval [31], [32]. To shed some light on how significant the difference is, we conducted a field test where three volunteers were asked to type a printed article in "Word" on the same laptop. Volunteers can finish the task at their own pace and methods (e.g., they can select and copy/paste words, use mouse to move the cursor back for correction, etc). However, volunteers are not informed about the goal of the test beforehand to avoid unnecessary interference. First, we analyzed the intervals between typing "i" and "s" ("is" appeared 238 times in the article) and the CDF is shown in Fig. 1f. We can see that the distributions of the intra-key intervals from three volunteers vary by 55%. Similarly, the CDF of the typing speeds (i.e., number of characters typed every 5 seconds as shown in Fig. 1g) implies the users can have significantly different habits while typing.

Are EM patterns consistent for the same user?

We asked the three volunteers to repeat the same test for 3 times in different days. The CDF of intra-key intervals and typing speeds are shown in Fig. 1f and 1g, respectively. We can see that the distributions of the same volunteer across different days are similar.

Summarize above, the EM signals caused by user-device interactions were shown to different across users and remain consistent over time, which suggests that EM signals could be used for user fingerprinting.

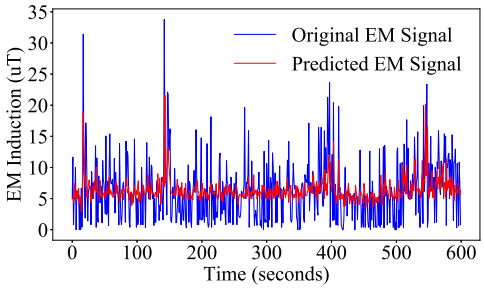


Fig. 3: EM interference prediction using 2-layer LSTM model. The blue line represents the original EM signals from background Apps and the red line represents the predicted EM signals.

IV. MagPrint System Design

In the following section, we outline the design of the proposed MagPrint system and user fingerprinting algorithm.

A. System Overview

Fig. 2 illustrates the system architecture of MagPrint. The system comprises four main units: data collection, interference cancellation, App category identification, and user fingerprinting. Raw electromagnetic (EM) signals are collected by a magnetic sensor (note that it could be an external device as our prototype or a built-in sensor in mobile devices). The signals are then pre-processed using a cancellation scheme and various filters for the removal of interference. Since user characteristics vary according to the nature of the App, classifiers are used to categorize the Apps in order to mitigate these effects. Finally, an algorithm performs feature extraction and classification for use in establishing user fingerprints.

B. Interference Cancellation

Cancelling EM signals from background Apps: As shown in Fig. 10a, EM signals decay exponentially with distance. Since the magnitude of EM signals generated by user-device interactions is around $40\mu T$, environmental EM interference has a negligible effect on the user fingerprint. Therefore, EM interference is due primarily to the internal operations of mobile devices. Extracting clean patterns pertaining to the operation of specific Apps by specific users requires the removal irrelevant interference.

Our experiment results revealed that EM interference varies over time; however, changes in the signals tend to be gradual. This means that it should be possible to train a model for the prediction and removal of background interference from raw signals.

To cancel the interference, received EM signals are first divided into sliding windows to facilitate analysis. Thus, the EM signals can be represented as: $X = [X_1^d, X_2^d, ...X_l^d]$, where X_i^d is a 1D vector representing the time-series of the EM signal in a window the length of which is measured in d milliseconds. For a given continuous EM signal collected over a period of time X, our task is to predict signal X_k^d where $k > l$. We developed a simple 2-layer LSTM (long-short term memory network, to mention later.) model with 32-units and soft-max loss functionality for prediction. As shown in Fig. 3, the RMSE (root-mean-square error) loss of the testing process reached 0.35, which suggests that the signals are of sufficient stability to allow the use of a simple LSTM model for interference prediction and cancellation.

TABLE I: Differences in user interactions with Apps in seven categories. The frequencies of typing, moving, and clicking are indicated on a scale of 1-5, where 1 refers to the lowest frequency and 5 refers to the highest frequency.

Frequency of	Typing	Clicking	Moving
Internet	3	5	5
Business	5	5	3
Communication	5	3	3
Game	1	3	5
Multimedia	1	1	1
SNS	3	3	5
System	3	4	3

Filtering EM noise: MagPrint then filters out other forms of EM noise that are not related to user operations (e.g., high-frequency noise and random noise). The frequency range of signals related to most user operations overlap very little with other high-frequency signals from Apps. This means they are easily differentiated by applying a low-pass filter specific to that frequency range (0Hz - 1000Hz). A Gaussian filter is then used to eliminate random noise in order to obtain EM signals specific to the user-device interactions.

C. App Category Identification

The characteristics of user-device interaction patterns tend to vary according to the nature of the Apps. For example, interactions with videos mostly involve the mouse, whereas those with document programs mostly involve the keyboard. This makes it difficult to create an universal fingerprint model (i.e., applicable to all Apps). Furthermore, the enormous variety of Apps that are currently in use would also make it impossible to train an individual model for each one.

To address the issue, we found that many different Apps actually present similarities in terms of usage characteristics. For example, using Microsoft Word and Power Point exhibit similar typing characteristics. The observation makes it possible to apply a single fingerprint model for multiple similar Apps. As shown in Table I, we adopted the method outlined in [12] for the categorization of Apps based on the frequency of typing, cursor movement, and clicking. As a result, we can resolve the issue by identifying the App categories according to usage characteristics before trying to establish a user fingerprint.

To identify the App category, we employed a Random Forest (RF) classifier. For given EM signals after interference cancellation, we extract frequency domain data and feed these features into a RF classifier. As shown in Fig. 13, RF classification achieved average accuracy of 97.0% when applied to identify the categories of 30 various Apps.

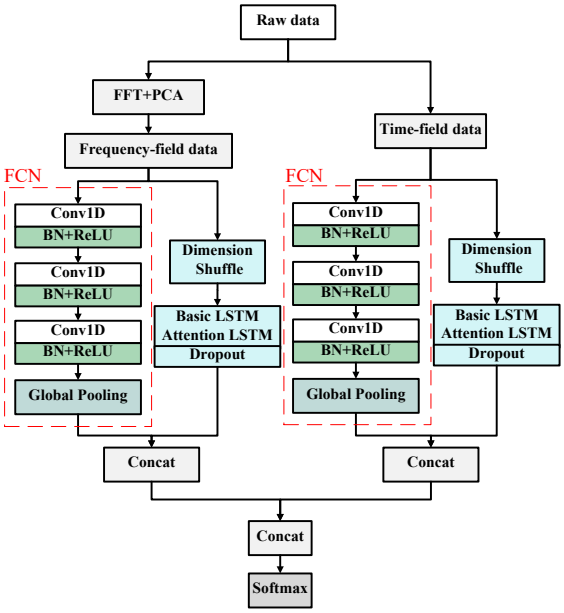


Fig. 4: *TF-FCN-LSTM* structure

D. User Fingerprinting Algorithm

After recognizing App category, we propose our algorithm *TF-FCN-LSTM* for fingerprinting users. Data normalization is first used to mediate the effects of data scale [33]. The data are then divided into sliding windows, whereupon data pertaining to the time and frequency domain of each window are fed into the *TF-FCN-LSTM* network for feature extraction and classification. Each of these steps is detailed below.

1) Feature Extraction with FCN: Most conventional approaches to feature extraction are time-consuming and require specific domain knowledge [34]. Thus, we employed a deep learning algorithm to automate the extraction of features. Fully convolutional networks (FCN) have been show to perform well in the extraction of features from time-series data as it can accept input signals of any size and then retain the spatial information after all convolutional operations [**?**]; therefore, we adopted this approach for the extraction of features in the temporal and frequency domains. We applied the fast Fourier transform (FFT) and principal component analysis (PCA) to time-domain data in each window to compute data in the frequency domain, as follows:

$$f_i^d = PCA(FFT(X_i^d)) \tag{1}$$

As shown in Fig. 4, data in the time and frequency domains are fed with the classification results from the upper layer into a *TF-FCN-LSTM* functional block for feature extraction and classification. Fig. 4 presents details of each *TF-FCN-LSTM* block. The red dashed box in Fig. 4 shows that the three components (i.e., convolution blocks) of the FCN are stacked for feature extraction. Each component includes a convolutional layer followed by batch normalization (BN) to accelerate convergence and avoid over-fitting, and a ReLU as the activation function. From the convolution blocks, the features are fed into a global average pooling layer rather than a fully connected layer in order to reduce the number of weights [35].

Specifically, the above operations above can be described as:

$$\begin{aligned} Conv_k(Y) &= b_{i,k} + W_{i,k}^d \times Y \\ FCN_k(Y) &= \text{ReLU}(\text{BN}(Conv_k(Y))) \end{aligned} \tag{2}$$

where $Conv_k(Y)$ represents the output of the k-th convolutional layer (in the time or frequency domain), Y are used as the input. $FCN_k(Y)$ represents the features after FCN. Combining the input time series (in the time and frequency domains from Eq. 1, we obtain the following:

$$\begin{aligned} T_i^d &= FCN(X_i^d) \\ F_i^d &= FCN(f_i^d) \end{aligned} \tag{3}$$

where T_i^d and F_i^d represents the features in the time and frequency domains after FCN.

2) Enhance Feature Extraction via Triplet Loss: One of the main challenge to use EM signals for user fingerprinting is that the EM signals associated with a given user tend to change over time and may even vary with their mood. In order to extract a feature set which is robust against these dynamics require a complex network structure and a large training dataset.

In this work, we proposed to use triplet loss [36], [37] to address the challenge. We used triplet loss to guide the training of FCNs in order to create a new feature space in which the distances between EM data from a given user are small, and the distances between EM data from different users are large.

To be more specific, suppose we have three EM signals as input: anchor signal I^a, positive instance signal I^p, and negative signal I^n, where I^a and I^p, is from the same user, whereas I^n is from another user. Then signal I is then translated to a d-dimension embedding $f(I)$ following this formula:

$$||f(I^a) - f(I^p)||_2^2 + \alpha < ||f(I^a) - f(I^n)||_2^2 \tag{4}$$

and the corresponding loss function can be written as:

$$L = max(||f(I^a) - f(I^p)||_2^2 - ||f(I^a) - f(I^n)||_2^2 + \alpha, 0) \tag{5}$$

where α is a margin that controls the distance between the positive pair (I^a, I^p) and negative pair(I^a, I^n)). Adding the triplet loss function to the CNN makes it possible to map the input features into a new space in order to aggregate EM data from one user, and separate data from other users in order to enhance the representativeness of FCN.

3) Classification Model: Conventional time-series classification methods are unable to retain behavior characteristics covering a long time span, thus we adopted the Long Short-Term Memory Network (LSTM) [27] in conjunction with the proposed TF-FCN (Time-Frequency Fully Convolutional

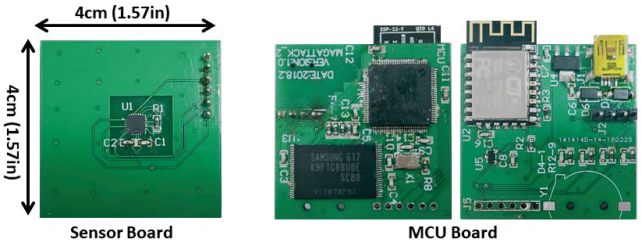


Fig. 5: Prototype designed to collect EM signals

Network) to leverage the features from the time domain as well as the frequency domain.

As shown in the blue dashed line box of the Fig. 4, temporal field data $t_i(X_i^d)$ and frequency field features f_i^d are first passed into the dimension shuffle block which can reduce the computation time of training without reducing accuracy. Then they are passed into the LSTM block which comprises of a basic LSTM layer and an Attention LSTM layer followed by a dropout. The basic LSTM layer can learn feature pattern among different time steps, while the Attention LSTM can selectively learn input signals and improves accuracy when the sequence is very long. The output of the global pooling layer and the LSTM block are concatenated and passed into a softmax classification layer to compute the loss. The procedure can be summarized as follows:

$$T_1 = \text{Concat}(LSTM(t_i), T_i) \quad (6)$$

$$T_2 = \text{Concat}(LSTM(f_i), F_i) \quad (7)$$

$$All_i = \text{Concat}(T1, T2)) \quad (8)$$

$$Out_i = \text{Softmax}(All_i), \quad (9)$$

where T_1 and T_2 are temporary output of LSTM network, All_i is the output of network, and Out_i is the final output of the *TF-FCN-LSTM* following an activation function.

V. Prototype

In this section, we describe the implementation of the prototype and the proposed data collection methods. Electromagnetic signals from the MCU and Wi-Fi modules in the prototype would interfere with EM data from the mobile device. Thus, we designed the prototype with two discrete circuit boards (a sensor board and an MCU board), as shown in Fig. 5. During data collection, the sensor board is attached directly to the mobile device in order to obtain clear EM signals, while the MCU board is kept at a distance from the sensor board in order to prevent interference.

The single sensor in the sensor board (DRV425; Texas Instruments) is a fluxgate magnet sensor, which is used to sense the magnitude of single-axis magnetic fields. The analog output of the sensor is transmitted via pins to the MCU board (STM32F407) to digitize the signal and control the embedded system. An RTC module (DS1307) is used to record the data samples with timestamps, and a Wi-Fi module (ESP8266) enables the wireless uploading of sensor data to the server.

TABLE II: List of 30 Apps collected in the experiments.

App Category	Apps
Internet	Chrome, Firefox, Internet Explorer, Amazon Shopping, Baidu Cloud Download
Business	Microsoft Word, Excel, Power-point, Microsoft Notepad, Adobe Acrobat XI Pro
Communication	Skype, Tencent WeChat, QQ
Game	Zuma, Candy Crush Saga, Minecraft, Plants vs. Zombies, Agar Online
Multi Media	Youtube, Tencent Video, Aqiyi Video, Potplayer, NetEase cloud Music, Windows Media Player
SNS	Gmail, Github, Twitter
System	System Player, System Camera, System 3-D Plot

TABLE III: List of 10 devices collected in the experiments.

Model	OS versions	CPU Speed(GHZ)
MacBook Air MQD32CH/A	MacOS 10.13	1.7
MacBook Pro MMGM2CH/A	MacOS 10.13	2.8
Hp ENVY14-J102TX	Windows 10	1.6
Hp 15-be101TX	Windows 10	2.5
Lenovo T440	Windows 10	2.4
ASUS Vivobook 4000	Windows 10	2.4
ASUS FX-PRO	Windows 8	2.4
Samsung 800G5M-X08	Windows 8	2.5
Dell Ins-15PD-7745BR	Ubuntu 17.10	2.3
Acer SF314-52-59TW	Ubuntu 17.10	2.5

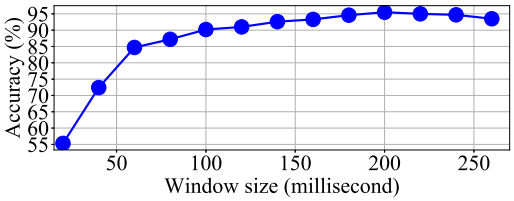
In other words, the DRV425 sensor board outputs the analog signals to the STM32F407 MCU board, which digitizes the signals (using a 12-bit ADC at a sampling rate of $10KHz$) and then instructs the Wi-Fi module to transmit time-series EM data to a remote server. After denoising and normalization, the server sends the preprocessed data to the trained classifier to perform user fingerprinting.

VI. Evaluation

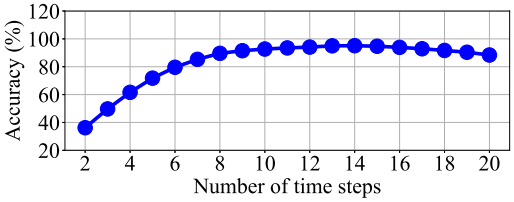
A. Experiment Setup

Data Collection: Our prototype was designed to collect electro-magnetic (EM) data in real time at a sampling rate of 10KHz. A 100Gb EM dataset was collected from 30 volunteers (25 males and 5 females aged 20-45 years) using 30 Apps on 10 differently laptops. All data were manually labeled with the information pertinent to the user, App, and device. Each volunteer used each App on each device for at least 10 minutes. As shown in Table II, we referenced [12] for the selection of representative Apps. Note that several commonly-used Apps were selected for each of the categories. Table III lists the devices used in the experiment; i.e., 10 types of laptop with a range of CPU speeds and 4 OSes.

Evaluation Metrics: For each user, we collect his/her EM signals when operating on different Apps and devices of different times. Thus we have multiple records of the same user. We then split these different records using a 10-fold cross-validation, and combine records of all volunteers for training and testing. The average accuracy of the 10-fold multi-classification are reported as the evaluation metric for user fingerprinting. Similarly, we can obtain the evaluation metric for classifying different Apps, App category by aggregating the corresponding data.

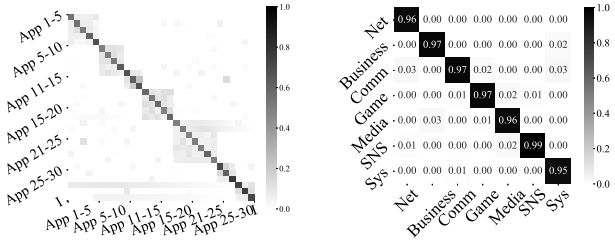


(a) Impact of window sizes



(b) Impact of time steps

Fig. 6: Impact of system parameters.



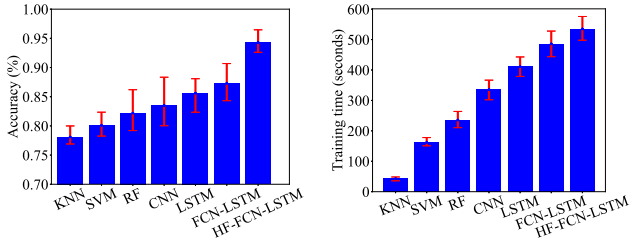
(a) App Classification (b) App Category Classification

Fig. 7: Performance of App category classification. (a) is confusion matrix of 30 Apps. The 3 Apps are listed in Table II following the same order. (b) represents classification result of the 7 App categories.

B. Micro-benchmark

We first evaluate the impact of system parameters and the performance of the system building units.

1) Deciding Hyper Parameters: The main parameters affecting *TF-FCN-LSTM* are the window size and the time step, which respectively determine the learning ability of the FCN model and LSTM model. A larger window provides more content for FCN, thereby facilitating the extraction of obvious features. More time steps give the LSTM model a longer input sequence for the extraction of transparent patterns. Nonetheless, optimizing these settings involves a compromise, due to the fact that too many time steps or a window that is too large would result in a small sample data scale, which would no doubt compromise accuracy. During training, we first set the time step length to 20 (default) in order to determine the window size (see Fig. 6a). We obtained an optimal value when the window reached $200ms$. This is because typing or moving frequency of humans are less than $100Hz$ in common case, so $200ms$ of time window is enough to learn short interval operation features. We then fixed the window at $200ms$ to select the number of time steps. We obtained an optimal value at approximately 12 (see Fig. 6b).



(a) Classification accuracy. (b) Computation time.

Fig. 8: Evaluation of classifiers. (a) and (b) compare classification accuracy and running time of different algorithms.

2) Evaluation of App Category Identification: Since user interaction characteristics on different Apps vary greatly, we need to infer the App type before fingerprinting users. Fig. 7a presents the confusion matrix of directly classifying 30 Apps using a Random Forest classifier, and the average accuracy is 83%. The result is reasonable as many Apps may emit similar EM patterns when they are working, such as Chrome and Firefox. Meanwhile, Fig. 7a also show that Apps of the same category are mixed up, while different categories are isolated. As discussed in Table I, similar user interaction characteristics are exposed on Apps from the same App category, which suggests that classifying App category is enough. The App category classification results are shown in Fig. 7b. We achieved the average accuracy of 97.0% on 7 App categories.

3) Evaluation of TF-FCN-LSTM *:* The proposed *TF-FCN-LSTM* scheme was compared with six existing machine learning methods: KNN and SVM (based on universal distance), Random Forest (a popular ensemble learning algorithm), classical CNN and LSTM models, and FCN-LSTM (the state-of-the-art approach to classify time-series data). We combined the data obtained from each user for all Apps and all devices, and then averaged the user classification results using 10-fold cross validation. As shown in Fig. 8a, *TF-FCN-LSTM* achieved the best results in the classification of users, with an average accuracy of 94.3%. The conventional machine learning methods failed to exceed accuracy of 85%. This is a clear demonstration that *TF-FCN-LSTM* can address the challenges while using EM patterns for user fingerprinting and outperform the state-of-the-art time-series classification method by 10.9%.

4) Computation Efficiency of TF-FCN-LSTM *:* All of the experiments were run on our server (32 CPU cores) equipped with a high-performance graphic card (Tesla K80). Fig. 8b presents the training time of *TF-FCN-LSTM* and the other baseline algorithms. The scale of the historical data was very large; therefore, pre-processing required a lot of time. KNN and RF were the easiest models to train, due to the fact that they lack the complex network structures of deep learning models. Among the deep learning models, the classical LSTM required the least time; however, *TF-FCN-LSTM* greatly increased classification accuracy with only a 5% time penalty compared to classic LSTM. This is because the special network structure enables *TF-FCN-LSTM* to using FCN and LSTM

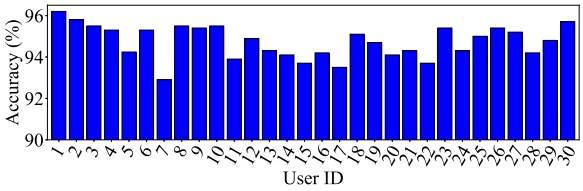


Fig. 9: User fingerprinting results of 30 users using 30 Apps on 10 different mobile devices.

TABLE IV: Variations in classification accuracy with the number of devices, Apps, and users

#Device	Acc
2	100%
3	98.8%
5	97.4%
8	96.1%
9	95.4%
10	94.3%

# App	Acc
5	98.9%
10	98.1%
15	97.4%
20	96.7%
25	95.4%
30	94.3%

#User	Acc
5	98.8%
10	98.4%
15	98.3%
20	97.3%
25	95.9%
30	94.3%

model in parallel.

C. Overall Performance

We then evaluate the performance of MAGPRINT in user fingerprinting and the impact of various factors to MAGPRINT.

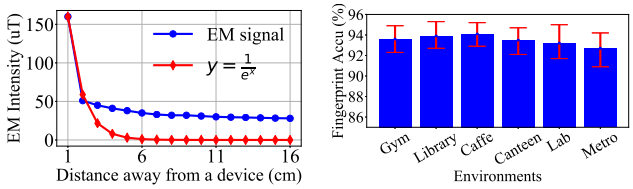
1) Evaluation of User Fingerprinting: Fig. 9 shows the accuracy of MAGPRINT in fingerprinting users when 30 users were using 30 Apps on 10 different mobile devices. This is a clear demonstration of the efficacy of the proposed scheme in identifying users. The accuracy for each individual ranges from 92.9% to 96.2% and is 94.3% in average.

2) Impact of Number of Devices, Apps, and Users: We then evaluate the impact of various dynamics to the system to show the robustness of MAGPRINT. As shown in Table IV, even after increasing the number of users, MAGPRINT was able to achieve classification accuracy of 94.3%. Table IV also shows the variation in average accuracy with the number of devices or Apps. These results were obtained by averaging the data from one user using all of the Apps or devices via 10-fold cross validation.

3) Impact of Environment: According to [38], the intensity of electromagnetic wave decays as it travels through conductive material, which can be expressed as:

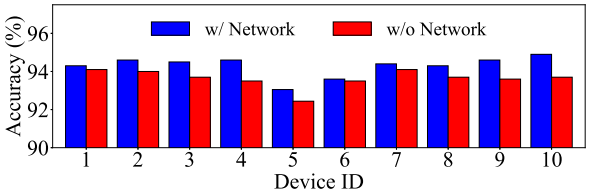
$$\delta = \sqrt{\frac{2}{\mu_0 \varepsilon_0}} \frac{1}{\omega(\sqrt{1+(\frac{\sigma}{\omega\varepsilon_0}-1)^{1/2}})} \simeq \sqrt{\frac{2}{\mu_0 \sigma \omega}}$$

where δ represents the "skin depth", defined as the depth below the surface of the conductor at which the current density has fallen to $1/e$ (approximately 0.37). ϵ_0 denotes the electric permittivity of free space, and μ_0 represents the magnetic permeability of free space, σ is a constant indicating the electrical conductivity, and $\omega = 2\pi \times f$, f is the frequency of EM signals. According to the formula, EM signals attenuate exponentially with the distance traveled, with the result that the effects of the, and this feature makes the effect of surrounding

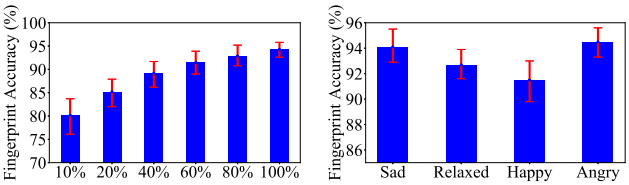


(a) EM signals decay over distance. (b) Environments

Fig. 10: Impact of sensing distance and environment.



(a) Network connection



(b) Battery states. (c) Users' mood.

Fig. 11: Impact of device states (network connection and battery states) and users' mood.

electromagnetic environment are extremely small. As shown in Fig. 10a, the magnetic probe in the experiment was placed at various distances in front of a strong magnetic device to record changes in EM intensity at various distances. Our results demonstrate that the decay effects were exponential. Furthermore, at distances exceeding $1cm$, there was an 83% reduction in EM intensity, which suggests that the surrounding EM environments would have no effect on the signals used for user fingerprinting. Due to the rapid signal decay over distance and the available space for the sensor was less than $5cm$, we opted to attach the sensor directly to the mobile devices.

To further evaluate the impact of EM interference under a variety of environmental conditions, we conducted experiments in gymnasiums, a library, a cafe, a canteen, and our laboratory where there are many computers. The results are shown in Fig. 10b. The fact that the average fingerprinting accuracy was 94% proves that the proposed system was unaffected by nearby EM sources.

4) Impact of Device States: We also evaluated a number of factors that vary with device usage, such as battery states and whether there are network connections. The results in Fig. 11b and Fig. 11a show that though network connections make little difference to the fingerprint result, the battery left can indeed affect the performance of MAGPRINT. This is due to the fact that when the battery power is low, the device automatically eases the execution of CPU instructions to protect the battery, which is reflected in the EM signals.

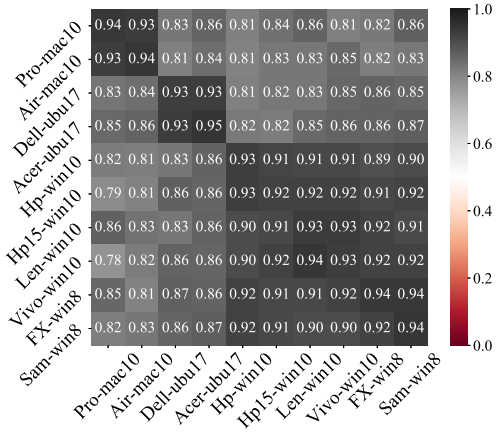


Fig. 12: Leave-one-device-out cross validation.

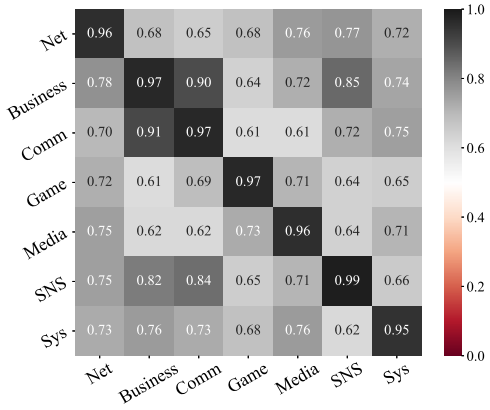


Fig. 13: Leave-one-category-out cross validation.

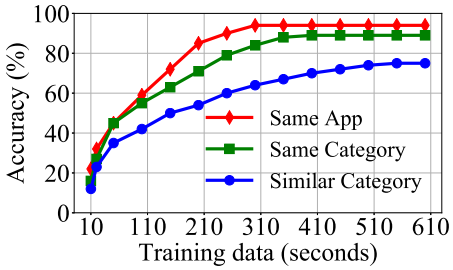


Fig. 14: Evaluation of sensing distance and data scale. The figure presents the amount of data from a previous unseen user required for training on a given App. Note that the figure compares data from the same category or a similar category.

5) Impact of User States: The mood of the user can also affect the way that they use their Apps. Excited users might type more quickly and sleepy users might make more mistakes and need to use the mouse more for corrections. As shown in Fig. 11c, we used models that were pre-trained under normal conditions to fingerprint a single user while happy, sad, angry, and relaxed [39]. Our results indicate that the mood of the user can indeed influence performance; however, the accuracy is still over 92%, which is within the range deemed acceptable. The results suggested the effectiveness of adopting the triplet loss to extract features which are independent to the user states.

6) Leave-One-Device-Out Performance: We evaluate the performance of the leave-one-device-out cross validation. That is, we use 9 of 10 devices to train a model and test it with EM signals collected on the 10^{th} device. Despite considerable variations among mobile devices in terms of CPU and GPU operations, the EM signals generated by the memory card, hard disk, electronic fan, and battery tend to be stable, and therefore easy to eliminate. This makes it possible for classification models trained for one device to be used for other devices. Fig. 12 illustrates the behavior of the proposed system when performing leave-one-device-out cross validation. MAGPRINT achieved average accuracy of 87.5%. Furthermore, we observed that the degree of accuracy was closely related to the OS types. In other words, sharing models among devices with the same OS (92.0%) was easier than migrating models across OSes (83.7%).

7) Leave-One-Category-Out Performance: We use EM signals collected from 29 out of 30 users to train a model and test data collected on the 30^{th} user. Similar to leave-one-device-out cross validation, it's also possible to adapt user classification models trained on one App category to another category that unseen in the training dataset. Fig. 13 shows the confusion matrix of leave-one-category-out cross validation. Results show that when there are 7 different App categories, some App categories can adapt well to models that trained on others, with accuracy of 90%, while others can only achieve about 75%. This is because different App categories can reflect different dimensions of user characteristics. For example, users mainly use mouses when playing games, while they mostly type in "Word".

8) Data Scale Evaluation: The practicality of the MAGPRINT system depends on minimizing the number of training samples that are required. As mentioned earlier, recognizing users on a given App does not require historical data specific to that App; i.e., we require user behavior only for an App in the same category or a similar category. This greatly reduces the amount of data required for training. Fig. 14 illustrates the amount of data required to achieve acceptable performance. Overall, it appears that 5 minutes of training data of users running a given App is sufficient to achieve accuracy of 95%. In other scenarios where it is not possible to obtain data from precisely the same App, 10 minutes of data is required to achieve accuracy of 89% (using data from the same App category) or 75% (using data from a similar App category).

VII. CONCLUSION

In this study, we developed a novel user fingerprinting system using electromagnetic (EM) signals to characterize user-device interactions. We developed a novel deep learning-based classification model referred to as *TF-FCN-LSTM*, which uses EM signals for fingerprinting. We also developed a prototype of the proposed system (MAGPRINT) for the collection of EM signal data. The proposed device is easily attached to any

smart phone or laptop. Extensive experiments revealed that the proposed MAGPRINT outperforms current state-of-the-art methods in terms of identification accuracy.

MAGPRINT is a new concept, and considerable work is still required. In the future, we will compile an enormous huge training set that includes numerous devices, applications, and users to further improve identification accuracy. We will also work on reducing the computational overhead of the algorithm in order to reduce delays in identification tasks. It is also conceivable that MAGPRINT could be implemented in a wide range of application scenarios, such as energy saving and privacy for mobile devices.

ACKNOWLEDGEMENT

This work is supported by the Joint Key Project of NSFC (U1736207), National Key R&D Program of China (2017YFC0803700), and Startup Fund for Youngman Research at SJTU.

REFERENCES

[1] Gartner. Gartner forecasts flat worldwide device shipments until 2019. http://www.gartner.com/newsroom/id/3560517, 2018.

[2] shopify. 12 time management apps to organize your life and keep you on track. https://www.shopify.com/blog/time-management-apps/, 2019.

[3] AVC. Screen time tracking/management. https://avc.com/2019/01/screen-time-tracking-management/, 2019.

[4] Eric Malmi and Ingmar Weber. You are what apps you use: Demographic prediction based on user's apps. In *Tenth International AAAI Conference on Web and Social Media*, 2016.

[5] H. Carter P. Marquardt, A. Verma and P. Traynor. (sp)iphone: Decoding vibrations from nearby keyboards using mobile phone accelerometers. *Conference on Computer and Communications Security – CCS 2011*, pages 551–562, 2011.

[6] S. Katzenbeisser S. Biedermann and J. Szefer. A study on power side channels on mobile devices. *Symposium of Internetware – Internetware 2015*, pages 30–38, 2015.

[7] S. Zhang T. Zhu, Q. Ma and Y. Liu. Context-free attacks using keyboard acoustic emanations. *Conference on Computer and Communications Security – CCS 2014*, page 453–464, 2014.

[8] Grunge. Kids who wasted thousands of dollars on gaming. https://www.grunge.com/29299/, 2017.

[9] CBCNews. Pembroke parent gets 8k$ xbox bill after son racks up charges. https://www.cbc.ca/news/canada/ottawa/pembroke-xbox-bill-8000-1.3397534/, 2019.

[10] CSOonline. Former employee visits cloud and steals company data. https://www.csoonline.com/article/3265109, 2018.

[11] Itarchiteks. Data theft from employees. https://www.itarchiteks.com/preventing-data-theft-employees/, 2014.

[12] Hengshu Zhu, Enhong Chen, Hui Xiong, Huanhuan Cao, and Jilei Tian. Mobile app classification with enriched contextual information. *IEEE Transactions on mobile computing*, 13(7):1550–1563, 2014.

[13] X. Chen L. Yan, Y. Guo and H. Mei. Hard drive side-channel attacks using smartphone magnetic field sensors. *Financial Cryptography – FC 2015*, 8975:489–496, 2015.

[14] Z. Ba K. Ren C. Zhou C. Song, F. Lin and W. Xu. My smartphone knows what you print: Exploring smartphone-based side-channel attacks against 3d printers. *Conference on Computer and Com- munications Security – CCS 2016*, page 895–907, 2016.

[15] G. Zhou A. Farajidavar Q. Yang, P. Gasti and K. S. Balagani. On inferring browsing activity on smartphones via usb power analy- sis side-channel. *IEEE Trans. Information Forensics and Security*, pages 1–1, 2016.

[16] Daniel Genkin, Itamar Pipman, and Eran Tromer. Get your hands off my laptop: Physical side-channel key-extraction attacks on pcs. *Journal of Cryptographic Engineering*, 5(2):95–112, 2015.

[17] Daniel Genkin, Lev Pachmanov, Itamar Pipman, and Eran Tromer. Stealing keys from pcs using a radio: Cheap electromagnetic attacks on windowed exponentiation. *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 207–228, 2015.

[18] Edward J Wang, Tien-Jui Lee, Alex Mariakakis, Mayank Goel, Sidhant Gupta, and Shwetak N Patel. Magnifisense: Inferring device interaction using wrist-worn passive magneto-inductive sensors. *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 15–26, 2015.

[19] Mustafa Gokce Baydogan, George Runger, and Eugene Tuv. A bag-of-features framework to classify time series. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2796–2802, 2013.

[20] Wanpracha Art Chaovalitwongse, Ya-Ju Fan, and Rajesh C Sachdeo. On the time series k-nearest neighbor classification of abnormal brain activity. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 37(6):1005–1016, 2007.

[21] Young-Seon Jeong, Myong K Jeong, and Olufemi A Omitaomu. Weighted dynamic time warping for time series classification. *Pattern Recognition*, 44(9):2231–2240, 2011.

[22] Jason Lines and Anthony Bagnall. Time series classification with ensembles of elastic distance measures. *Data Mining and Knowledge Discovery*, 29(3):565–592, 2015.

[23] Zhicheng Cui, Wenlin Chen, and Yixin Chen. Multi-scale convolutional neural networks for time series classification. *arXiv preprint arXiv:1603.06995*, 2016.

[24] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Lstm-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148*, 2016.

[25] Zhiguang Wang, Weizhong Yan, and Tim Oates. Time series classification from scratch with deep neural networks: A strong baseline. In *Neural Networks (IJCNN), 2017 International Joint Conference on*, pages 1578–1585. IEEE, 2017.

[26] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[27] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[28] John Cristian Borges Gamboa. Deep learning for time-series analysis. *arXiv preprint arXiv:1701.01887*, 2017.

[29] Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Shun Chen. Lstm fully convolutional networks for time series classification. *IEEE Access*, 6:1662–1669, 2018.

[30] Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Samuel Harford. Multivariate lstm-fcns for time series classification. *Neural Networks*, 116:237–245, 2019.

[31] Kyle O Bailey, James S Okolica, and Gilbert L Peterson. User identification and authentication using multi-modal behavioral biometrics. *Computers & Security*, 43:77–89, 2014.

[32] Jun Chen, Guang Zhu, Jin Yang, Qingshen Jing, Peng Bai, Weiqing Yang, Xuewei Qi, Yuanjie Su, and Zhong Lin Wang. Personalized keystroke dynamics for self-powered human–machine interfacing. *ACS nano*, 9(1):105–116, 2015.

[33] Guang Deng and LW Cahill. An adaptive gaussian filter for noise reduction and edge detection. In *Nuclear Science Symposium and Medical Imaging Conference, 1993., 1993 IEEE Conference Record.*, pages 1615–1619. IEEE, 1993.

[34] Isabelle Guyon, Steve Gunn, Masoud Nikravesh, and Lofti A Zadeh. *Feature extraction: foundations and applications*, volume 207. Springer, 2008.

[35] Gunnar Schmidtmann, Graeme J Kennedy, Harry S Orbach, and Gunter Loffler. Non-linear global pooling in the discrimination of circular and non-circular shapes. *Vision Research*, 62:44–56, 2012.

[36] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[37] Xiao Zeng, Kai Cao, and Mi Zhang. Mobiledeeppill: A small-footprint mobile deep learning system for recognizing unconstrained pill images. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, pages 56–67. ACM, 2017.

[38] Constantine A Balanis. *Advanced engineering electromagnetics*. John Wiley & Sons, 1999.

[39] Xiao Hu and J. Stephen Downie. When lyrics outperform audio for music mood classification: A feature analysis. In *ISMIR*, 2010.