JOURNAL OF INFORMATION SCIENCE AND ENGINEERING XX, XXXX-XXXX (2024)
DOI: XX.XXX.XXXX.XX

Lane Identification Using Wireless Sniffing and Deep Learning for Intelligent Transportation Systems

WEI-HSUN LEE, TSUNG-HSIEN LIU AND CHE-YU CHANG+
Department of Transportation and Communication Management Science
National Cheng Kung University
No. 1, University Road, Tainan 701, Taiwan, R.O.C.
E-mail: leews@mail.ncku.edu.tw; hans.th.liu@gmail.com; r58091018@gs.ncku.edu.tw+

Wireless sniffing technology (WST) has emerged as a promising approach for traffic data collection in intelligent transportation systems (ITS). This technique enables the capture of critical traffic metrics such as travel speed, travel paths, and origin-destination (OD) data. Despite its advantages, challenges such as lane identification (LIP), handling multiple devices (MDP), and transportation mode classification (TMP) remain unresolved. In this study, multivariate time-series data were gathered using X-topology WST sniffers, and four deep learning models—LSTM, Encoder, Inception, and ResNet—were employed to address the LIP challenge. The primary obstacles include short contact windows and fluctuating signal strength. To mitigate these issues, Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT) were applied, effectively smoothing signal variations and improving accuracy under short contact window conditions. The experimental results demonstrate that integrating FFT/IFFT improves the F1-score by approximately 1–9 percentage points compared to models without this technique, particularly under short contact conditions. This approach significantly reduces reliance on lengthy and costly data collection efforts, increasing both robustness and cost-efficiency. Field trials highlight ResNet's superior performance with an F1 score of 0.972, marking an 18% improvement over traditional machine learning models. However, due to its high computational demands, LSTM is recommended for scenarios with limited resources or budget constraints.

Keywords: Wireless Sniffing Technique, Lane Identification Problem, Time Series Classification, Deep Learning, Connected Vehicle

1. INTRODUCTION

Real-time traffic information plays a critical role in modern intelligent transportation systems (ITS). Traditionally, vehicle detector (VD), automatic vehicle identification (AVI), and GPS positioning-based vehicle probing (GVP) [1] are the major traffic information data sources. Electronic toll collection (ETC) based vehicle probing (EVP), Cellular-based Vehicle Probing (CVP) [2], and AI-based traffic flow recognition [3] are the emerging traffic information collection techniques. Vehicular network and connected vehicle techniques enabled by DSRC and C-V2X (5G) can gather real-time and accurate

Received ; revised ; accepted.
+Corresponding author

dynamic vehicle information, however the penetration rate of DSRC-enabled (or C-V2X enabled) vehicles as well as the infrastructure are still very low.

With the growing use of mobile devices in vehicles, such as Android Auto, Apple CarPlay, Bluetooth handset or speaker, in vehicle Wi-Fi access point, etc., an increasing amount of in-vehicle wireless signals can be collected through the roadside equipment. Recently, a novel traffic information collection technique is proposed, which applies wireless sniffing technique (WST) to collect traffic information [4-8]. WST is a cost efficient, real-time and non-intrusive traffic monitoring scheme [5, 8], the roadside unit sniffs wireless signals emitted from in-vehicle mobile devices which contains the unique hardware number such as media access control address (MAC) of the mobile device and the signal radio strength represented by Received signal strength indication (RSSI). Combining with local roadside unit information, including GPS position and timestamp, the roadside unit can analyze the collected information to get real-time traffic information, such as speed, travel path, origin-destination (OD), intersection turning ratio, etc.

Although traffic information gathered by WST is promising since it has the advantages of being cost-effective, real-time, and non-intrusive, three problems remain to be solved because there is a gap between mobile device information and traffic information. These problems are the lane identification problem (LIP), which identifies the lane position of the vehicle; the multiple devices problem (MDP), which determines the accurate count of vehicles instead of the mobile devices; and the transportation mode problem (TMP), which identifies the type of transportation mode accurately [9]. In [9], these problems are defined, and machine learning models are applied to tackle them. Among these problems, LIP is important because traditional traffic data sensors, GVP/EVP/CVP, cannot resolve LIP effectively. With accurate lane information, more precise lane-based traffic management actions can be performed to reduce traffic congestion and improve traffic network performance.

In [10], a hierarchical support vector machine (SVM) is applied to determine which lane the vehicle is located in, the experiment result shows that the accuracy of identifying whether the vehicle changes lanes is about 60%-65% and the accuracy of identifying which lane the vehicle is located in is 79%-82% in the no change lane scenario. However, some issues remain in this study that make it impractical. First, the accuracy of the identifying lane position is not good enough for practical application; Second, most machine learning models need a long contact window (more than 20 seconds) to achieve an acceptable performance; however, it is difficult to gather such a long contact window in the highway scenario. Third, many influencing factors such as radio interference, brand or types of mobile devices make large variations in received signal strength data so that many models or algorithms based on the linear signal variation assumption is not applicable.

The Wireless Sniffing Technique (WST) is applied to a bi-directional, four-lane scenario, as shown in Fig. 1. A set of WST sensors is deployed in specific topologies, such as the *X-type* $\{S_1, S_2, S_3, S_4, S_5\}$, the *rectangle type* $\{S_1, S_3, S_4, S_5\}$, or the *diamond type* $\{S_2, S_4, S_5, S_6\}$, to collect wireless signals emitted from moving vehicles. The four lanes in two opposite directions are labeled as $[R_1, R_0, G_0, G_1]$. The *Lane Identification Problem (LIP)* is defined as determining which lane a moving vehicle occupies based on the collected raw wireless signal data. When a vehicle equipped with a mobile device emits wireless signals, it is detected by the WST sensors as it passes through the field. The

collected raw data are represented as four-tuples:

(Sensor#, MAC#, RSSI, Timestamp).

In this study, the X-type topology is adopted as a performance-oriented benchmark configuration. This choice is based on previous WST-based lane identification studies, in which the X-type topology showed better lane-identification performance than the rectangle-type and diamond-type topologies. In addition, the five-sensor X-type configuration provides richer multivariate RSSI time-series observations from both longitudinal and lateral sensor positions. Such spatially diverse observations are beneficial for deep learning models to learn the spatiotemporal signal patterns associated with different lanes. However, this study does not claim that the X-type topology is the cost-optimal deployment strategy.

Formally, the LIP is defined in Eq. (1). Given a vehicle equipped with a mobile device M_i that emits wireless signals (e.g., Wi-Fi hotspot sharing or Bluetooth communication), the LIP can be formulated as a classification problem that determines the lane label based on the time series of received signal strength (RSSI) data collected by the deployed sensor topology.

$$M_i \in \{R_1, R_0, G_0, G_1\} \tag{1}$$

In the literature, several related works were devoted to generating traffic information by using WST. Ding et al evaluated wireless signal data and developed a data mining algorithm to extract traffic information, including speed, travel time and traffic volume, and the experiment results reveal the feasibility of applying Wi-Fi sniffing technique to collect real-time traffic information [7]. Chen et al proposed intelligent traffic monitoring using wireless signals; they designed a transmitter and receiver link of Wi-Fi modules across the road to gather the Wi-Fi channel state information. A convolutional neural network (CNN) is applied for vehicle classification with an accuracy of 87.3% [4]. However, both Chen, et al. [4] and Ding, et al. [7] do not propose solutions for the LIP.

For the lane identification problem, Rompis, et al. [11] developed a lane identification methodology for GPS-based probing vehicles for estimating the queue dynamics on intersections with multiple lanes. Won, et al. [5] proposed a solution by using Wi-Fi channel state information, which is similar to Chen, et al. [4]. Experiment results show that it achieved about 90% accuracy in both urban road and highway on lane detection accuracy. Fan [6] identified the vehicle lane position in a no lane-changing long tunnel scenario by a voting scheme with Z-type sensor topology; however, many sensors are needed for the voting scheme, which is not cost-effective. Lee, et al. [9] applied the hierarchical support vector machine to identify lane-changing facts first, and then determine which lane the vehicle is in. Experiment results show that the X-type topology outperforms rectangle-type and diamond-type topologies, and the accuracy on the fixed lane scenario is 79% - 82%, however, several issues still need to be improved as discussed above.

To apply LIP solution to a practical intelligent transportation system, four key issues must be properly solved. First one is the short contact time issue, the accuracy of the proposed solution should good enough under smaller WST contact window, for example, under 15 seconds because it is difficult for the sensor to keep long contact period with

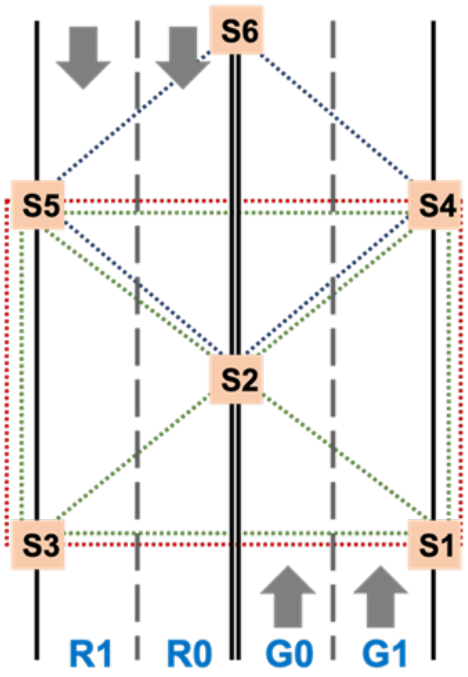


Fig. 1. The concept of WST on collecting wireless signal data from vehicles.

mobile device in high speed scenario due to the radio signal coverage constraint; Second one is the signal variation challenge, the proposed model should be able to handle large amount of wireless signal raw data with high variations; Third one is the robust issue that the accuracy of the proposed solution should be good and stable. Fourth, the real-time issue requires that lane-identification results be generated within a short response time for practical ITS applications. In this study, the real-time requirement is evaluated primarily from the model inference perspective, while the complete end-to-end latency of an online WST processing pipeline is discussed as an implementation issue.

In this paper, LIP is formulated and solutions that tackle the four key issues mentioned above are proposed. Taking the signal variation challenge as an example, Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT) are applied to filter the noise of the signal and smooth the time series of the wireless signal raw data. Four deep learning neural networks are implemented and compared to determine which one is best suitable for solving LIP, which includes long short-term memory (LSTM) [12], Encoder [13], Inception [14], and ResNet [15]. The major contributions of this paper are summarized as fourfold.

1. For the LIP, this study shows that multivariate time series raw data collected by X-type topology WST sniffers can be applied to practical ITS applications. It can achieve high lane-identification performance with fast model inference time (less than 30 ms per batch for a 100-step time series, batch size = 128), outperforming

the compared models [5, 10].

2. FFT/IFFT is applied in the proposed solution, which is proven to be effective for short contact period data. The F1-score is improved by approximately 1–9 percentage points compared with models without FFT/IFFT when the contact period is less than 15 seconds. This enhances robustness and cost-effectiveness since long-term contact data collection is unstable and high-cost.

3. Four deep learning models (Inception, ResNet, Encoder, and LSTM) are designed and compared for solving the LIP. Among them, ResNet outperforms the others, achieving the highest F1-score of 0.972. For short contact periods of 3–15 seconds, the average F1-score is 0.842, and it increases to 0.966 when the contact time window lasts 18–39 seconds.

4. A k-fold blocked cross-validation scheme is designed to prevent testing data from being monitored during model training, ensuring that the models are fairly validated.

The remainder of this paper is organized as follows. In DATA PREPROCESSING, the problem description of LIP, data cleaning and preprocessing are presented, and the FFT/IFFT is applied to filter the noise of the signal and smooth the time series of the wireless signal raw data. DEEP LEARNING MODELS presents mathematic model of four deep learning neural networks applied in this work, including LSTM, Encoder, Inception and ResNet. Next, the experiment results and discussions are presented. Finally, the conclusion remarks and some future directions are discussed.

2. DATA PREPROCESSING

2.1 Field Dataset Description

The field dataset used in this study was collected from a WST-based lane-identification field experiment on Chengnan Road, Annan District, Tainan City, Taiwan. The site was a semi-enclosed bidirectional road section with two lanes in each direction. During the experiment, multiple Wi-Fi-enabled mobile devices were placed in a passenger car, and the vehicle passed through the sensing area several times at approximately 30 km/h. Five Wi-Fi sniffers were deployed in an X-type topology to collect RSSI signals. The lane labels were defined as R1, R0, G0, and G1, corresponding to the outer and inner lanes in the two travel directions.

For the lane-identification task, the raw RSSI records were grouped by MAC-address identifier and vehicle pass, and then reorganized as five-variate time-series samples. The non-uniform raw sampling intervals were resampled at 0.3-second intervals. After trip segmentation, lane labeling, resampling, and data cleaning, 314 valid MAC-address-level trip samples were obtained. Each sample corresponds to one detected MAC-address identifier within one labeled vehicle pass, rather than to one physical vehicle trip. Different time-window lengths from 10 to 130 steps were evaluated, corresponding to contact windows of 3 to 39 seconds.

Device information was organized mainly at the MAC-address level. Detailed metadata such as device brand, chipset manufacturer, operating system version, antenna characteristics, and weather condition were not systematically recorded. Therefore, this study does not conduct a device-brand-specific, chipset-specific, or weather-specific sensitivity analysis.

2.2 Notations and Data Encoding

A vehicle with one or more mobile devices in active driving through the WST area with I sensors that collect the WST radio raw data, is defined as the notation of $[x^i_{(j,1)}, x^i_{(j,2)}, \ldots, x^i_{(j,t)}]$, where i denote the sensor number, j denotes the journey number, t denotes the t-th step timestamp with time period is τ_k, between each step. The value of $x^i_{(j,t)}$ will be assigned as -99 if the null is collected which indicates the vehicle may be too far from the WST sensors. The RSSI raw data collected from Wi-Fi sensors is defined as in Eq. (2), which indicates the N step time series data collected by sensor i from the timestamp m. The N-steps 5-variates time series data collected by X-type topology WST sensors can be presented as Eq. (3). The list, $\{(\mathrm{MTS}_1, y_1), (\mathrm{MTS}_2, y_2), \ldots, (\mathrm{MTS}_n, y_n)\}$, represents the LIP data source collected in n trips, y_i and is the labeled lane number. In Eq. (4), it presents the one-hot encoding of the classification result in each journey j, where the sequence y_i indicates the classification result of which lane the vehicle is located in.

$$RSSI^i_j = \left[x^i_{j,m}, x^i_{j,m+1}, \ldots, x^i_{j,m+N-1}\right]^{\mathrm{T}} \in R^{N\times 1}, \quad -99 \le x^i_{j,k} \le 0 \tag{2}$$

$$MTS_j = \left[RSSI^1_j, RSSI^2_j, RSSI^3_j, RSSI^4_j, RSSI^5_j\right] \in R^{N\times 5} \tag{3}$$

$$y_j = \left[y^j_1, y^j_2, y^j_3, y^j_4\right], \quad y^j_k \in (0,1) \tag{4}$$

2.3 Radio signal variation challenge

The basic idea of traffic information generation from the WST is that it analyzes the spatiotemporal relationships among WST sensors and moving vehicles (mobile devices). However, a very strong assumption is made that assumes the mobile device radio signal is ideally collected between the vehicle and WST sensors. The collected RSSI curve is expected to be bell-shaped, and it is inversely proportional to the distance from the moving vehicle, as illustrated in Fig. 2, where the number on the curve indicates the WST sensor number labeled in Fig. 1.

However, several problems must be faced due to the radio signal variations. First, the environmental effect on radio signal is dramatic, even the set of RSSI samples collected from the same device at a fixed location obtained over time, which will change the distribution of statistical properties [16]. Second, the RSSI collected varies greatly depending on the chipset manufacturers, even in the same distance case. Third, the time intervals of the sampling data are very unstable, ranging from 0.1 to 4 seconds. An example of a collected WST signal is shown in Fig. 3, which presents the signal variation challenge compared to the ideal case shown in Fig. 2.

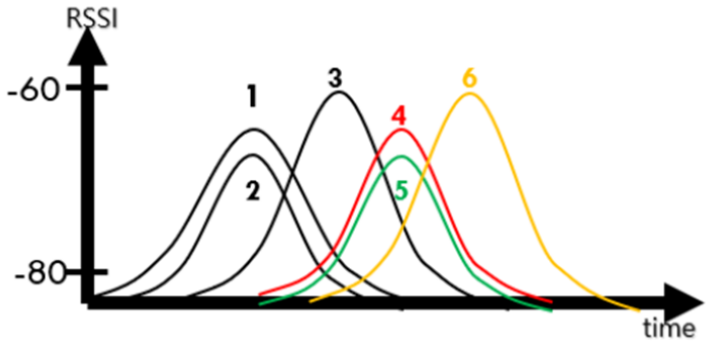


Fig. 2. Concept of ideal spatiotemporal signal collection under topologies.

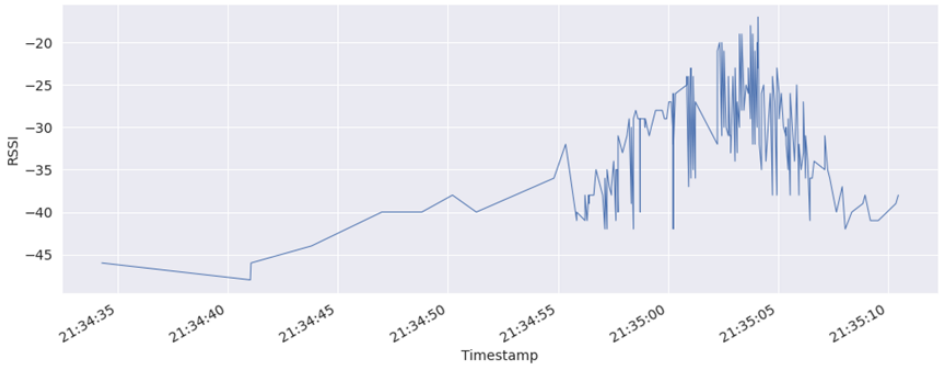


Fig. 3. An example of WST signal collected by a Wi-Fi sniffer (in RSSI).

2.4 WST signal preprocessing

The flow of WST signal data preprocessing is illustrated in Fig. 4, which consists of three subsystems. The upper part is the three data cleaning modules, including "trip segmentation and labeling", "time window moving average", and "missing values imputation". Two kinds of datasets are prepared for evaluation by k-fold validation ($k = 5$ in this paper), one is the raw dataset, averaging samples in fixed time intervals (300 ms), and the other is the smoothing dataset filtered by FFT/IFFT.

In practice, instead of using the full trip RSSI data collected, if we can use a shorter segment of the trip RSSI data, which shortens the length of MTS_j, to reach an acceptable accuracy of LIP, the robustness of the system can be enhanced. To evaluate the effectiveness and robustness of the proposed LIP classifiers for the best deployment decision, the length of different time windows (N-steps) is experimented with, where N ranges from 10 steps to 130 steps (3 to 39 seconds).

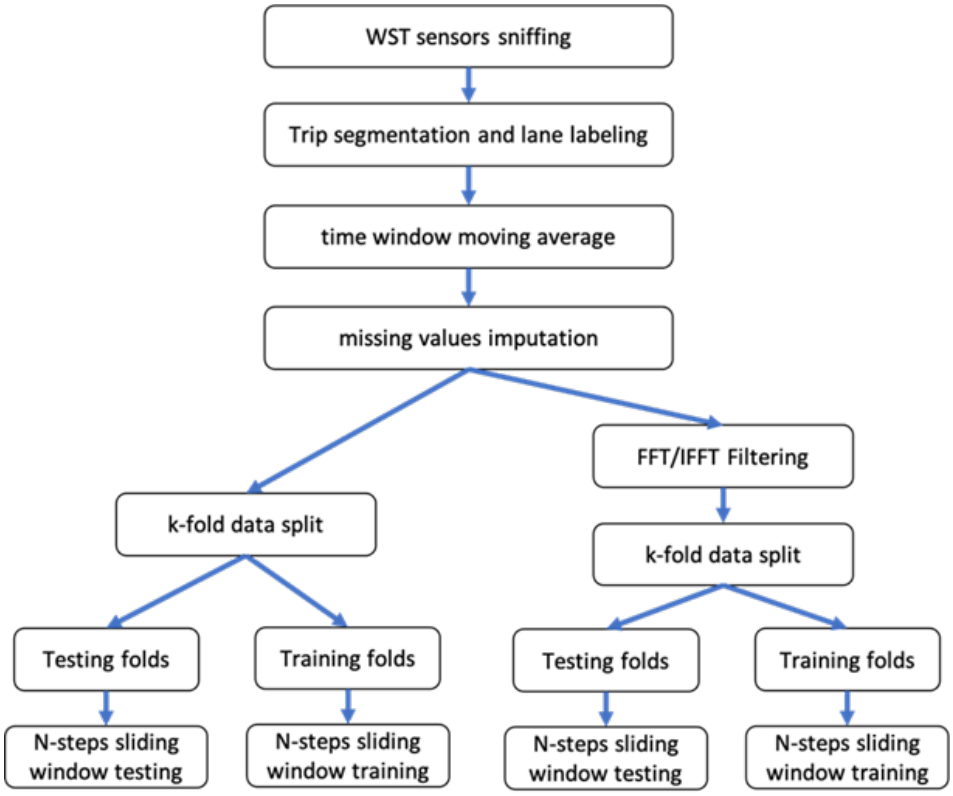


Fig. 4. WST signal data preprocessing and FFT/IFFT smoothing.

Most models that deal with time series problems require identical time intervals, that is, $\tau_k = \tau$. The time intervals of two adjacent RSSI values collected by the same sniffer range from 100 ms to 4 seconds, and the average time interval is close to 300 ms. In

the time window moving average module, $\tau = 0.3$ seconds is applied as the time series interval. If there are multiple samples within 0.3 seconds, we average these RSSI values to reduce the short-term variation. The machine learning models require the dataset without missing data; however, there are nearly 4% missing values after the sample data is averaged by the interval of $\tau = 0.3$ second. By observation, the collected Wi-Fi RSSI raw data forms time series that possess strong seasonality. From the experimental results, the Seasonally Decomposed Missing Value Imputation yields the best performance. We use the R package `imputeTS` function to impute the missing data, which forms the preprocessed time series raw dataset. In Fig. 5, it shows a multivariate time series of RSSI collected from five WST sensors in X-type topology of one go-trip on the inner lane, which is labeled as "G0". Note that the value on the y-axis will be -99 to indicate the vehicle is away from the sniffing range of the sensors.

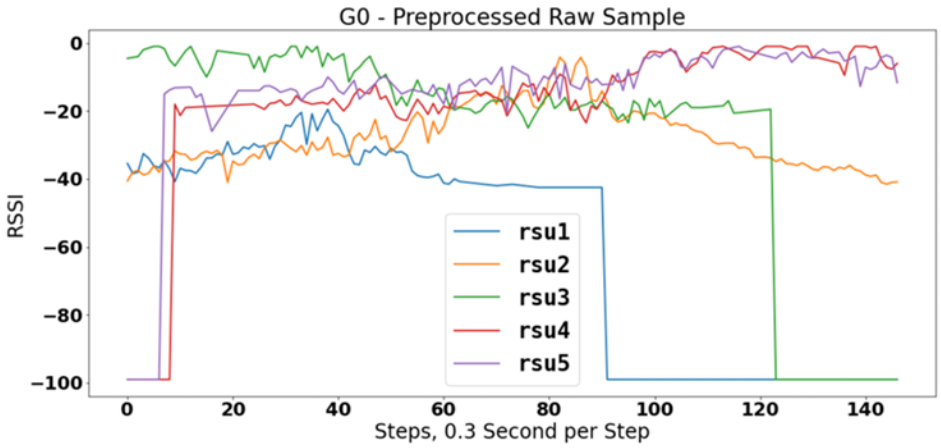


Fig. 5. Preprocessed multivariate time series sample.

2.5 Data smoothing by FFT / IFFT filter

In the field of communication signal processing, the Fast Fourier method is usually used to reduce noise interference, which inspires the idea of WST signal data smoothing. We try to use FFT/IFFT to filter the noise of the signal and smooth the time series raw data. FFT is a mathematical function that calculates the discrete Fourier transform of an input sequence. After FFT, we can get the spectra domain to identify the peaks in the FFT frequency magnitude chart. A threshold noise filter can be applied to accept those frequencies where the magnitudes exceed the threshold of the given quantile. After FFT filtering out the frequencies below the obtained quantile, IFFT can be applied to create the smooth dataset, as shown in Fig. 6.

2.6 K-Fold cross validation and Metrics

Due to the time series data serial correlation in temporal, the related application is often omitted in favor of an out-of-sample evaluation [17]. We can get many time series samples by sliding the time window in one trip; they are data-dependent within these

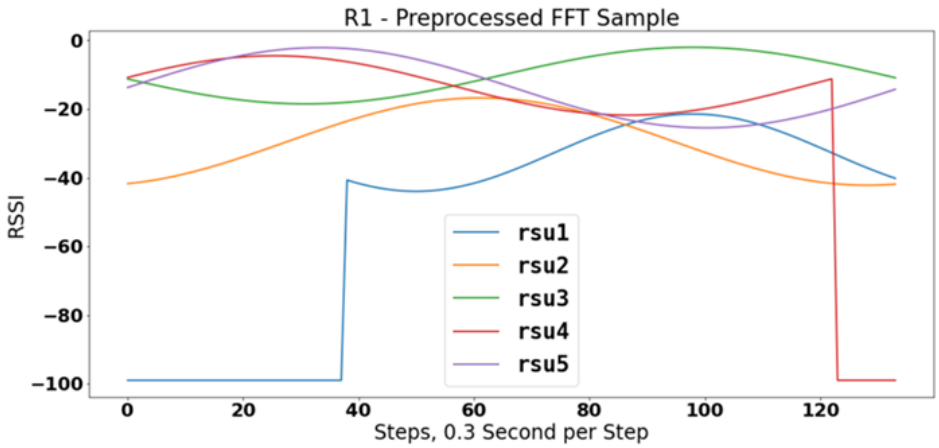


Fig. 6. A sample of applying FFT/IFFT for smoothing raw dataset.

samples, so we cannot use out-of-sample methods, which hold out the last part of the time series for testing, to validate the model. To address the overfitting or selection bias issues, cross-validation is applied to test the ability to predict new data that did not appear in the training phase. However, data dependencies are often observed in the time series data; these dependencies raise some caveats about using standard cross-validation in such data [18]. Several variants of k-fold cross-validation schemes have been proposed, specially designed for dependent data such as time series [17,18,19].

In this study, we proposed a revised k-fold blocked cross-validation in order to avoid these caveats. We split the training and testing dataset first, and slide windows respectively to sample training and testing time series. The main concern is the generalization ability of the model, i.e., its performance on unseen data. After the five-Fold splitting dataset, let Fold-k ($k = 1,2,\ldots,5$) be the testing dataset. The training dataset will be the other folds except for Fold-k. Let deep model C be the candidate model; we choose the best F1-score $F1_k^C$ to be the score of model C after training 13 times on the training dataset. Fig. 7 shows the accuracy of 5-fold cross-validation using the ResNet model, where the index in the X-axis indicates which fold is applied for testing and the other folds are applied for training. The performance index of Model C is defined in Eq. (5), and the goal of this paper is to find a classifier C, to maximize $F1_C$, as in Eq. (6).

$$F1_C = \frac{1}{5}\sum_{k=1}^{5} F1_k^C \tag{5}$$

$$F1_C = \arg\max_C F1_C \tag{6}$$

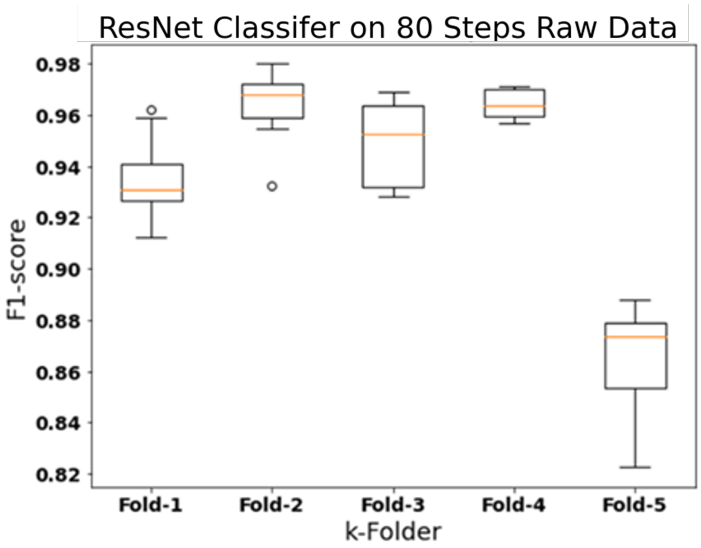


Fig. 7. The F1-score boxplot of five-fold cross validation on the ResNet model.

3. DEEP LEARNING MODELS

This section provides a detailed explanation of the proposed method, including the problem formulation and the models employed: LSTM, Encoder, Inception, and ResNet.

3.1 Problem formulation

For the n trips vehicles moving through the WST sensing area, the collected data set is formatted as

$$DS = \{(\mathrm{MTS}_1, y_1), (\mathrm{MTS}_2, y_2), \ldots, (\mathrm{MTS}_n, y_n)\},$$

where $\hat{y}_j = C_w(\mathrm{MTS}_j) = [\hat{y}_1^j, \hat{y}_2^j, \hat{y}_3^j, \hat{y}_4^j]$ is the probability distribution corresponding to the lane labels in ('R1', 'R0', 'G0', 'G1'). The goal is to find out a deep learning classifier model C and its w parameters set to predict the lane label of trip j (MTS_j).

The cost function would impact the training process and performance of deep learning models. In this study, the dataset collected is balanced, and the categorical cross-entropy is the most used loss function in DNNs [20]. We define the categorical cross-entropy loss function in Eq. (7), where $\mathscr{L}_w^j$ denotes the loss of sample j predicted by classifier C in the w parameter set.

$$\mathscr{L}_w^j = -\sum_{k=1}^{4} y_k^j \log \hat{y}_k^j \tag{7}$$

The error cost, $\mathscr{L}_w = \sum_{j \in DS} \mathscr{L}_w^j$, will be the total loss that classifier C_w obtains when

classifying the whole training set of *DS*. We train the classifier model *C* by backpropagation to find a set of parameters w^* such that

$$\mathscr{L}_{w^*} = \arg\min_{w} \mathscr{L}_w \tag{8}$$

The classifier C_{w^*} with the minimum loss will be the suggested Lane Identifier model *C*. In this study, we compared four deep learning classifier models: one LSTM and three other one-dimensional convolutional neural network models (1D-CNN), including Encoder, Inception, and ResNet, to find the suitable deep model for LIP.

3.2 Models

LSTM is first introduced by Hochreiter and Schmidhuber [12]. There are many studies using the LSTM model to predict traffic information if the dataset is in time series. Zhao, et al. [21] proposed an LSTM network model for traffic forecast, which considers spatiotemporal correlations in traffic systems via a two-dimensional network. To predict short-term traffic flow, LSTM and gated recurrent units (GRU) neural network methods perform better than the autoregressive integrated moving average (ARIMA) model [22]. For LIP, we need to find suitable classifiers. The LSTM-CNN accurately classifies five driving behaviors by judging the driver's driving style [23]. In the short-term sequence ordered data type task, the LSTM model was a well-known candidate, so we adapted stacked LSTM as a baseline model.

In order to conquer the exceedingly long-term dependencies, Yang, et al. [24] proposed an improved approach in which traffic flow values with higher impacts are captured using the attention mechanism. The attention mechanism idea drives us to consider the 1D-CNN encoder model as a candidate. There are many advantages to using 1D-CNN models compared with LSTM. RNNs make predictions in a sequential manner where the predictions for later timesteps are decided by their predecessors to complete. On the other hand, 1D-CNN can be done in parallel since the same filter is used in each layer, which can reduce the training time. In order to capture the time series local patterns, 1D-CNN can flexibly change receptive field size by stacking more layers or increasing the filter size [20]. There are systematic studies of time series classification using 1D or temporal CNN models [25][26][20], and the state-of-the-art multivariate time series classification deep models emerged nowadays. Three 1D-CNN models including Encoder [13], Inception [14], and ResNet [15] are chosen in this study to be the candidates, and they use very different architectures, mainly inspired from the image domain, to promote the performance of the model.

3.2.1 Encoder

Although the Encoder model is not the top performance among the nine deep models evaluated by [26], the special design of an Encoder is formed of a convolutional neural network with temporal output summarized by a convolutional attention mechanism [27]. The multivariate time series data can be transformed into a compact, fixed-length representation [27]. If used properly, in traffic problems with a dataset of time series, we can use this technique in transfer learning. That is, it first trains on a sufficient number of samples of the dataset to obtain learned representations of samples and transfers to the

small dataset to promote the performance of the model. We choose Encoder to be one of the candidates to understand if this kind of attention mechanism meets the prediction requirement of LIP. The Encoder model applied in LIP is a standard convolutional network with a convolutional attention mechanism to summarize the temporal time-series data, as shown in Fig. 8, where a fully connected layer is used in the final stage to set the desired representation dimensionality [27]. The designed attention mechanism, for a single filter, is defined as $h = h \cdot a$, where "$\cdot$" denotes a dot product, h is the result of a single one-dimensional convolutional filter over a time-wise signal, and a is the time-wise attention vector (independent for each filter).

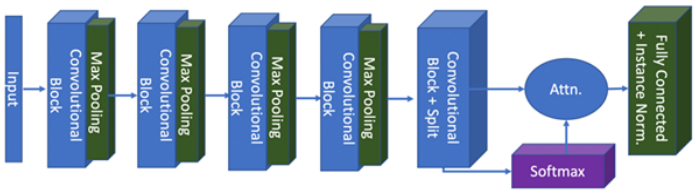


Fig. 8. Architecture of Encoder.

3.2.2 Inception

InceptionTime applies an ensemble of CNN models for a time series classification problem, which consists of five randomly initialized Inception networks [14]. The most specialized design is the Inception module, which is inspired by the vision deep model domain. The Inception network consists of three Inception modules, where the input of each residual block is transferred to the input of the next block via a shortcut connection to mitigate the problem of vanishing gradient [15].

3.2.3 ResNet

In Fawaz, et al. [26], nine deep learning neural network (DNN) models are studied and compared for time series classification, where ResNet outperforms other DNN models, including FCN, Encoder, MCNN, TWIESN, etc. ResNet is designed with a deep flexible architecture, which is much more successful than shallower architectures [15]. Compared with Encoder and Inception, ResNet focuses on stacking more convolution layers to make the model deeper. We choose ResNet as the candidate to make clear whether a deeper model is helpful for LIP. The shortcut residual connection between consecutive convolutional layers is applied in the ResNet, which differs from traditional convolutions in that a linear shortcut links the output of a residual block to its input. It enables the flow of the gradient directly through these connections [28] and makes training a DNN model easier by reducing the vanishing gradient effect [26,28]. The architecture of the applied ResNet DNN model is illustrated in Fig. 9, where the network is a combination of three residual blocks, a global pooling layer, and a softmax classifier.

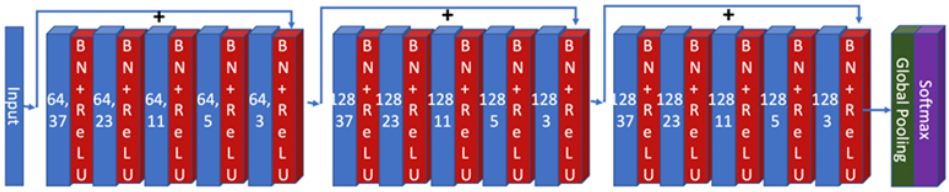


Fig. 9. The architecture of ResNet for solving LIP.

4. RESULTS AND DISCUSSIONS

The field experiment is arranged as shown in Fig. 1, where five WST sensors (S1, S2, S3, S4, S5) equipped with Wi-Fi sniffers are deployed as an X-type topology, the distance between (S1, S4) and (S3, S5) is 200 meters. With the Wi-Fi wireless communication range is at most 100 meters, the wireless signal coverage range is about 400 meters, where the signal contact time window would be about 36 seconds (120 steps) if the vehicle speed is 40 km/hr, 48 seconds (160 steps) if the vehicle speed is 30 km/hr.

To fairly compare the performance of LSTM and the three 1D-CNN models, we first evaluated one-, two-, and three-layer stacked LSTM architectures, where each LSTM layer contained 128 hidden units. As shown in Fig. 10, the F1-score of the one-layer LSTM is comparable to those of the two- and three-layer LSTM models. Therefore, the one-layer LSTM was selected as the baseline model.

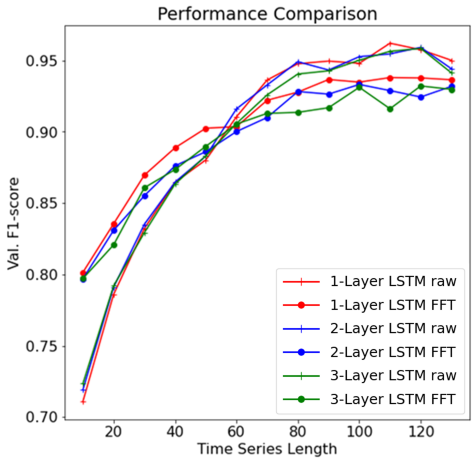


Fig. 10. Performance comparison with different numbers of stacked LSTM layers.

To improve reproducibility and ensure a fair comparison, the major hyperparameters used in the experiments are summarized as follows. For the LSTM model, one-, two-, and three-layer stacked architectures were evaluated, and each LSTM layer contained 128 hidden units. Since the one-layer LSTM achieved comparable performance with lower

complexity, it was selected as the baseline model. For the 1D-CNN-based models, including Encoder, Inception, and ResNet, a common receptive-field setting was adopted. The convolutional filter sizes were set to 37, 23, 11, 5, and 3. All models were trained using categorical cross-entropy loss, and early stopping was applied when the validation loss did not improve for 15 consecutive epochs. The models were evaluated using five-fold cross-validation under time-window lengths from 10 to 130 steps for both raw and FFT/IFFT-preprocessed datasets.

Following the five-fold blocked cross-validation scheme described in Section 2.6, each model was trained and evaluated under N-step time-series inputs, where N = 10, 20, ..., 130, using both raw and FFT/IFFT-preprocessed datasets. The comparison results of four DNN classifiers are shown in Fig. 11, where the results of both raw and FFT/IFFT datasets are presented. The result of Encoder under 50 steps is missed because of the adoption of five filter sizes 37, 23, 11, 5, and 3 for a large receptive field size. All 1D-CNN models adopt the same receptive field size for a fair comparison; this makes the Encoder model unsuitable for very short input sequences under the adopted receptive-field setting.

Compared with the baseline model (one-Layer LSTM), the ResNet outperforms others regardless of Raw or FFT/IFFT datasets, followed by Inception, LSTM, and Encoder. Under the adopted common receptive-field and training settings, the Encoder model showed lower performance than the other models. Therefore, the present results indicate that the specific Encoder configuration used in this study was less suitable for the LIP dataset than ResNet, Inception, and LSTM. However, this result should not be interpreted as a general conclusion that attention mechanisms are ineffective for LIP, since model-specific hyperparameter optimization was not exhaustively conducted. ResNet, the deepest model in this comparison, is composed of three residual blocks, and each block contains five convolution layers with filter sizes of 37, 23, 11, 5, and 3, respectively. It achieves a high F1-score of around 0.97 when the time-window length is greater than 70 steps on the raw dataset.

The result of the Raw and FFT/IFFT dataset comparison is presented in Fig. 11, where the results reveal the solution for the short contact period issue. For short time-window lengths, especially below 60 steps, FFT/IFFT smoothing is generally beneficial. The ResNet and Inception models get better results with an average F1-score of 0.842. When the time window is sufficiently long, the deep learning models can learn useful temporal patterns directly from the raw RSSI sequences. The accuracy gap between raw and FFT/IFFT data is around 0.04 because the FFT and IFFT process filters some information in time series fluctuation. For the case of a contact time window longer than 60 steps (18–39 seconds), the Raw dataset can achieve better accuracy with an average F1-score of 0.966 by using ResNet.

Table 1 summarizes representative F1-scores of the ResNet model under different time-window lengths. The results show that FFT/IFFT preprocessing improves the F1-score when the contact window is short, especially at 10 and 30 steps. However, when longer time windows are available, the raw data achieve higher F1-scores. This result supports our observation that FFT/IFFT is mainly beneficial for short-contact scenarios, while raw data are preferable when sufficient RSSI sequences can be collected. It should be noted that Table 1 reports representative observed F1-scores rather than statistical significance results. Therefore, the comparison should be interpreted as an empirical comparison under the adopted experimental settings. More rigorous statistical analysis with

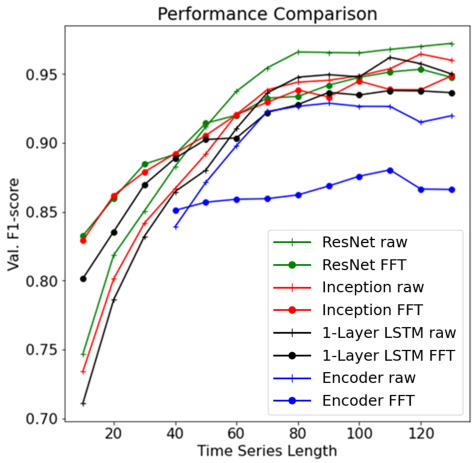


Fig. 11. Performance comparison of different deep learning models under raw and FFT/IFFT-preprocessed datasets.

Table 1. Representative F1-score comparison of ResNet under different time-window lengths.

Steps	Time (s)	Raw	FFT/IFFT	Difference (FFT/IFFT - Raw)
10	3	0.749	0.835	+0.086
30	9	0.852	0.886	+0.034
60	18	0.937	0.920	-0.017
90	27	0.967	0.941	-0.026
120	36	0.970	0.951	-0.019
130	39	0.974	0.946	-0.028

standard deviations, confidence intervals, and significance tests is left for future work.

In addition to the performance evaluation, the computing efficiency of these DNN classifiers is also evaluated. We monitor and collect the power draw every 500 ms of the dedicated GPU on training long-term Raw datasets with lengths of 100, 110, 120, and 130 time series steps to understand the power consumption differences between models. The comparison of power consumption and computing efficiency is shown in Fig. 12, where the power consumption based on batch sizes 64, 128, and 256 of four models is presented. The power consumption comparison is focused on GPU power draw because most calculations are performed on CUDA high-performance computing during the training process. After training is completed, the full power draw measured value in the violin plot is shown. For each model, the upper position means higher power consumption, and the larger area of a cluster means more time required to complete the training.

For each model, the larger the batch size, the higher the power draw (high position) but less time is spent to complete training (smaller area). In terms of energy consumption, there is a trade-off between time spent and power consumption. Take ResNet as an example, there are two clusters in the violin plot. The upper cluster represents the power consumption when the optimization process of backpropagation is running; because of

high-density computing and multiple sample sizes, its area is several times that of the lower cluster. The lower cluster represents the power consumption during prediction on testing data, and its lower position and smaller area indicate there is no need for optimization and less data size in the prediction stage. Although ResNet has the highest accuracy, it also consumes a lot of computing power. The ResNet consumes several times more power than the LSTM model in the training phase. Considering the requirements of low power consumption and acceptable accuracy, the one-Layer LSTM may be the recommended model for LIP. It should be noted that the reported execution time refers only to the model inference stage, not the complete end-to-end latency of an online WST system. The full pipeline would also include RSSI data aggregation, 0.3-second resampling, missing-value imputation, sliding-window construction, and optional FFT/IFFT smoothing. Since the present experiments were conducted using an offline field dataset, these preprocessing steps were not benchmarked as a real-time streaming pipeline. Therefore, the results demonstrate sub-second model inference, but they do not yet guarantee full end-to-end real-time performance for online deployment.

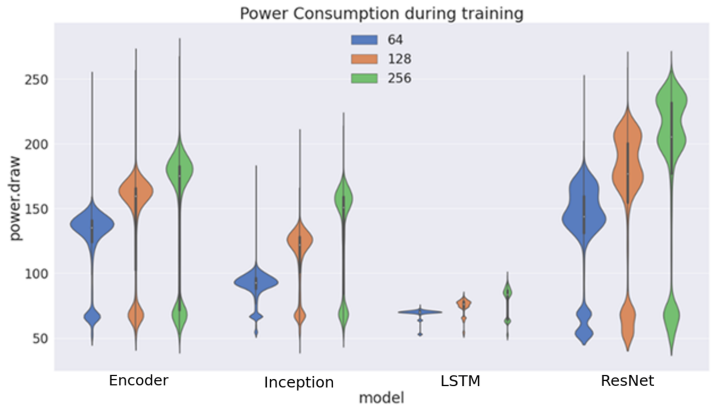


Fig. 12. Comparison of the power consumption and computing efficiency.

5. CONCLUSIONS

Applying the Wireless Sniffing Technique (WST) for collecting traffic information is one of the emerging approaches in intelligent transportation systems (ITS). It performs well in estimating travel time, vehicle speed, travel paths, and origin–destination pairs. However, several challenges remain in applying WST to the Lane Identification Problem (LIP), which make it still impractical for real-world traffic management systems. In this paper, we formulate LIP as a multivariate time series classification problem and implement four deep learning classifiers—LSTM, Encoder, Inception, and ResNet—to solve it. To address the long contact window and radio signal fluctuation issues, FFT/IFFT filters are designed to smooth the WST raw data. A k-fold blocked cross-validation scheme is

proposed to prevent testing data leakage during training, ensuring fair model evaluation. Five WST sensors are deployed in an X-type topology for field experiments. The results show that the ResNet model achieves the highest lane identification precision, with an accuracy of 97.3% and an F1-score of 0.972, representing an 18% improvement compared with the machine learning model in [10]. Moreover, under short contact windows (less than 50 steps or 15 s), models with the FFT/IFFT filter improve the F1-score by approximately 1–9 percentage points. These results demonstrate the effectiveness and robustness of the proposed model, particularly in highway environments where the wireless contact time window of WST sensors is typically less than 18 seconds.

Four key challenges in LIP—including short contact periods, signal variation, accuracy, and real-time responsiveness—are discussed and partially addressed in this study from the perspectives of preprocessing, model design, empirical performance, and model inference time. The contributions of this work are summarized fourfold as discussed in the INTRODUCTION. With the proposed WST solution, it is possible to collect travel time, travel paths, vehicle speed, and origin–destination data, as well as the lane position of vehicles for potential real-time ITS applications. Compared to other traffic information collection schemes such as EVP/CVP/GVP, the proposed method is more practical and cost-effective for real ITS applications and contributes to more precise, lane-based traffic management.

The suggestions for applying WST in practical ITS applications are summarized as follows:

1. FFT/IFFT filters should be applied when the wireless signal contact window is short (less than 18 seconds); otherwise, the raw data are recommended.

2. The ResNet model yields the highest accuracy but requires greater computing power, while LSTM is the fastest and most energy-efficient model, making it preferable when power or cost constraints exist.

For future work, although the ResNet model with the five-sensor X-type topology achieved promising lane-identification performance with an F1-score of 0.972, this topology should be regarded as a performance-oriented benchmark rather than a cost-optimal deployment strategy. Since deploying and maintaining five roadside sensors within a 200-meter corridor may increase implementation cost, future research will investigate simplified WST topologies, such as horizontally deployed two- or three-sensor configurations with shorter contact windows of less than 10 seconds. In addition, ensemble learning may be employed to combine WST data with varying sequence lengths to further enhance robustness and classification performance. A formal cost-benefit analysis should also be conducted by jointly considering lane-identification accuracy, sensor deployment cost, maintenance cost, computational cost, and operational feasibility. Furthermore, controlled sensitivity analyses under different device brands, chipset types, operating system versions, antenna characteristics, weather conditions, and roadside environments should be conducted to further evaluate the generalizability of the proposed WST-based lane identification framework.

ACKNOWLEDGMENT

This research was supported in part by the Ministry of Science and Technology under contracts MOST 109-2124-M-006-002, and "Center for Innovative FinTech Business Models" of National Cheng Kung University (NCKU), Taiwan, R.O.C.

REFERENCES

1. J. C. Herrera, D. B. Work, R. Herring, X. J. Ban, Q. Jacobson, and A. M. Bayen, "Evaluation of traffic data obtained via GPS-enabled mobile phones: The Mobile Century field experiment," *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 4, pp. 568–583, 2010.
2. D.-J. Lin, W.-L. Tsao, C.-H. Yu, H.-H. Liu, and Y.-C. Chang, "The Travel Time Prediction by Machine Learning Methods with Traffic Data in Chiayi City, Taiwan," in *Proc. 2019 4th Int. Conf. on Electromechanical Control Technology and Transportation (ICECTT)*, 2019, IEEE, pp. 257–260.
3. B. Tian, Q. Yao, Y. Gu, K. Wang, and Y. Li, "Video processing techniques for traffic flow monitoring: A survey," in *Proc. 2011 14th Int. IEEE Conf. on Intelligent Transportation Systems (ITSC)*, 2011, IEEE, pp. 1103–1108.
4. C. Chen, K. Ota, M. Dong, C. Yu, and H. Jin, "WITM: Intelligent Traffic Monitoring Using Fine-Grained Wireless Signal," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 3, pp. 206–215, 2020.
5. M. Won, S. Zhang, and S. H. Son, "WiTraffic: Low-cost and non-intrusive traffic monitoring system using WiFi," in *Proc. 2017 26th Int. Conf. on Computer Communication and Networks (ICCCN)*, 2017, IEEE, pp. 1–9.
6. Y.-H. Fan, "A Spatiotemporal Seamless Traffic Information Collection Framework by Intelligent Vehicle Probing," M.S. thesis, Dept. of Transportation and Communication Management Science, National Cheng Kung University, 2017.
7. F. Ding, X. Chen, S. He, G. Shou, Z. Zhang, and Y. Zhou, "Evaluation of a Wi-Fi Signal Based System for Freeway Traffic States Monitoring: An Exploratory Field Test," *Sensors (Basel)*, vol. 19, no. 2, Jan. 20, 2019.
8. M. R. Friesen and R. D. McLeod, "Bluetooth in intelligent transportation systems: a survey," *International Journal of Intelligent Transportation Systems Research*, vol. 13, no. 3, pp. 143–153, 2015.
9. W.-H. Lee, T.-J. Liang, and H.-C. Wang, "An innovative and cost-effective traffic information collection scheme using the wireless sniffing technique," *Vehicles*, vol. 4, no. 4, pp. 996–1011, 2022.
10. T.-J. Liang, "Applying Machine Learning to Analyze Vehicular Traffic Information Collected by Wireless Sniffing Technique," Master's thesis, Dept. of Transportation and Communication Management Science, National Cheng Kung University, 2019.
11. S. Y. Rompis, M. Cetin, and F. Habtemichael, "Probe vehicle lane identification for queue length estimation at intersections," *Journal of Intelligent Transportation Systems*, vol. 22, no. 1, pp. 10–25, 2018.
12. S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

13. S. Lange and M. Riedmiller, "Deep auto-encoder neural networks in reinforcement learning," in *Proc. 2010 Int. Joint Conf. on Neural Networks (IJCNN)*, 2010, IEEE, pp. 1–8.
14. H. I. Fawaz et al., "Inceptiontime: Finding AlexNet for time series classification," *Data Mining and Knowledge Discovery*, vol. 34, no. 6, pp. 1936–1962, 2020.
15. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
16. K. Kaemarungsi and P. Krishnamurthy, "Properties of indoor received signal strength for WLAN location fingerprinting," in *Proc. 1st Annu. Int. Conf. on Mobile and Ubiquitous Systems: Networking and Services (MOBIQUITOUS)*, 2004, IEEE, pp. 14–23.
17. C. Bergmeir, R. J. Hyndman, and B. Koo, "A note on the validity of cross-validation for evaluating autoregressive time series prediction," *Computational Statistics & Data Analysis*, vol. 120, pp. 70–83, 2018.
18. V. Cerqueira, L. Torgo, and I. Mozetič, "Evaluating time series forecasting models: An empirical study on performance estimation methods," *Machine Learning*, vol. 109, no. 11, pp. 1997–2028, 2020.
19. C. Bergmeir and J. M. Benítez, "On the use of cross-validation for time series predictor evaluation," *Information Sciences*, vol. 191, pp. 192–213, 2012.
20. S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.
21. Z. Zhao, W. Chen, X. Wu, P. C. Chen, and J. Liu, "LSTM network: a deep learning approach for short-term traffic forecast," *IET Intelligent Transport Systems*, vol. 11, no. 2, pp. 68–75, 2017.
22. R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," in *Proc. 2016 31st Youth Academic Annual Conf. of Chinese Association of Automation (YAC)*, 2016, IEEE, pp. 324–328.
23. S. Jia, F. Hui, S. Li, X. Zhao, and A. J. Khattak, "Long short-term memory and convolutional neural network for abnormal driving behaviour recognition," *IET Intelligent Transport Systems*, vol. 14, no. 5, pp. 306–312, 2019.
24. B. Yang, S. Sun, J. Li, X. Lin, and Y. Tian, "Traffic flow prediction using LSTM with feature enhancement," *Neurocomputing*, vol. 332, pp. 320–327, 2019.
25. Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in *Proc. 2017 Int. Joint Conf. on Neural Networks (IJCNN)*, 2017, IEEE, pp. 1578–1585.
26. H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: a review," *Data Mining and Knowledge Discovery*, vol. 33, no. 4, pp. 917–963, 2019.
27. J. Serrà, S. Pascual, and A. Karatzoglou, "Towards a Universal Neural Network Encoder for Time Series," in *Proc. CCIA*, 2018, pp. 120–129.
28. T. Ma et al., "CNN-based classification of fNIRS signals in motor imagery BCI system," *Journal of Neural Engineering*, vol. 18, no. 5, p. 056019, 2021.



Wei-Hsun Lee received the Ph.D. degree in computer science from National Chiao Tung University, Hsinchu, Taiwan, in 2009. From 1993 to 2010, he was an Associate Researcher with the Telecommunication Laboratories, Chunghwa Telecom Company Ltd., Taoyuan, Taiwan. He is currently a Professor with the Department of Transportation and Communication Management Science, National Cheng Kung University, Tainan, Taiwan. His research interests include vehicular ad hoc networks, spatiotemporal data mining, telematics systems, advanced traffic management systems, electronic toll collection systems, and near-field communication techniques.

PLACE PHOTO HERE

Tsung-Hsien Liu received the M.S. degree in Industrial Engineering and Engineering Management from National Tsing Hua University, Hsinchu, Taiwan. Photograph is not available at the time of publication.



Che-Yu Chang received the M.S. degree in Telecommunications Management from National Cheng Kung University, Taiwan. He is currently pursuing a Ph.D. degree with the Department of Transportation and Communication Management Science, National Cheng Kung University, Taiwan. His research interests include driving behavior analysis.