loxz digital

ShotSpotter Model v1.0

By Miu Lun (Andy) Lau, Chen Song
Data Scientist

Table of Contents

ABSTRACT

This model is the first model designed for Law Enforcement. It seeks to serve predicts future potential locations for gunfire. Using data from ShotSpotter API, this spatio-temporal model uses data from audio sensing equipment and immediately retrains the model to serve new locations where future gunshots will be fired. 3rd party demographic data, including latitude and longitude coordinates and time increments, help law enforcement determine how many units should be deployed to recommended locations. The training data for this Shotspotter model comes from the Chicago area but can be aligned with any city using Shotspotter. We are in the process of working with the UC Berkeley Police and the Oakland Police Department to deploy this model in production upon approval.

I . INTRODUCTION

Gun violence and gun crimes have done great social harms and it's difficult for police agencies to find out ahead of time specifically when and where will next gunshot happen. In this model, we implement convolutional long short term memory(LSTM) neural networks algorithm to predict time series of spatial dependencies on gunshots. This algorithm allows us to make real time predictions on the next potential gunshot location within next minutes or hours. In addition, we will also provide predictions on what types of gunshot that might happen next, whether it's single gunshot, multiple gunshots or it could be gunshot or fireplace. The demonstration of this model is to help police agencies to predict the location and time of future gunshots so that the police patrol and police protection can be distributed accordingly. Not only will police agencies benefit from the knowledge of the next potential gunshot, people can benefit from avoiding potential risky areas.

Ⅱ. DATA SETS

The preliminary dataset used in this model comes from shotspotter in the Washington DC area. It contains a total of 53758 independent events over a period of 7 years from 2014 to 2021. The key features include latitude and longitude coordinates and time increments, as well as the type of gunshot. Table I shows the used key attributes in metadata and the meaning of the attributes.

The model can be fit on the ShotSpotter alerts data from different police departments in different cities. And in the future, we consider to integrate other types of crime data into the model, such as robbery, murder, and burglary, etc. By integrating more crime data, the model will be able to provide predictions based on different crime types so that different police departments benefit from the model.

A. Data Engineering and Wangling

We removed all the irrelevant features in the raw data and the final dataset results in five data, from *UserID*, *Time*, *Longitude*, *Latitude*, and *Type of Gunshots*. *UserID* was sequentially generated manually because the raw data doesn't include the id. *UserID* is used for keeping track of the id of gunshot cases. The *Type of Gunshots* was converted to categorical feature. We use 0, 1, and 2 to denote single gunshot, multiple gunshot and multiple gunshot or firecracker individually. *Time*, *Longitude*, *Latitude* are all from the metadata.

Region Partition: There are many ways to divide a city into multiple regions in terms of different granularities and semantic meanings, such as road network [2] and zip code tabular [7]. In this work, we follow the previous work [12] to partition a city into $h \times w$ non-overlapping grid map based on the longitude and latitude.

B. Feature Engineering

We added *LocationID* as an unique identifier in the final dataset to identify unique location based on unique latitude-longitude pairs. We introduced bits calculation to generate LocationID. We shifted the float number latitude to the left by 16 bits places, then concatenated the bits format of latitude with the 16 unset bits format of the longitude. In this way, we created the *LocationID* as an unique identifier of gunshot location and feed the feature in the model.

To perform a Heat maps related studied, we first have to section off our dataset into different areas. We employed to use section off our dataset into (x,y) size of (36,36). For our dataset located in DC, each grid is approximately around (446 x 582) m or 259 km^2. The key features include latitude and longitude round (446 x 582) m or 259 km^2de coordinates and time increments, as well as the type of gunshot.

Ⅲ. TOOLS REQUIRED

We are using a number of software and packages used frequently by the data science community. For machine learning, we are using Keras [1]. Other common used packages is also utilizes for data parsing and visualizations such as **Numpy, Matplotlib, Pandas,** and **Scipy. Ipywidgets** is used for demonstration and selection of targets and other parameters. **BOTO3** is used for accessing and interacting with S3 data storage. **Plotly** [4] is used creating interactive figures to inferences with the APIs. To display existing events in google maps related data, we utilize **Gmaps** to **Yandex** to embedded google maps in notebooks.

Metadata	No. of Emails
Time	<str> UTC Time for occurring events
Longitude	<float> Longitude value for occurring event
Latitude	<float> Latitude value for occurring event
Location ID	<int> Unique identifier for each specific location
Gunshot_Type	<int> Types of Gunshots: • Single Gun-shot, • Multiple Gunshot, Multiple Gun-shot or Firecracke

Table 1. Final Data-set metadata and the corresponding value for campaign type and industry type.

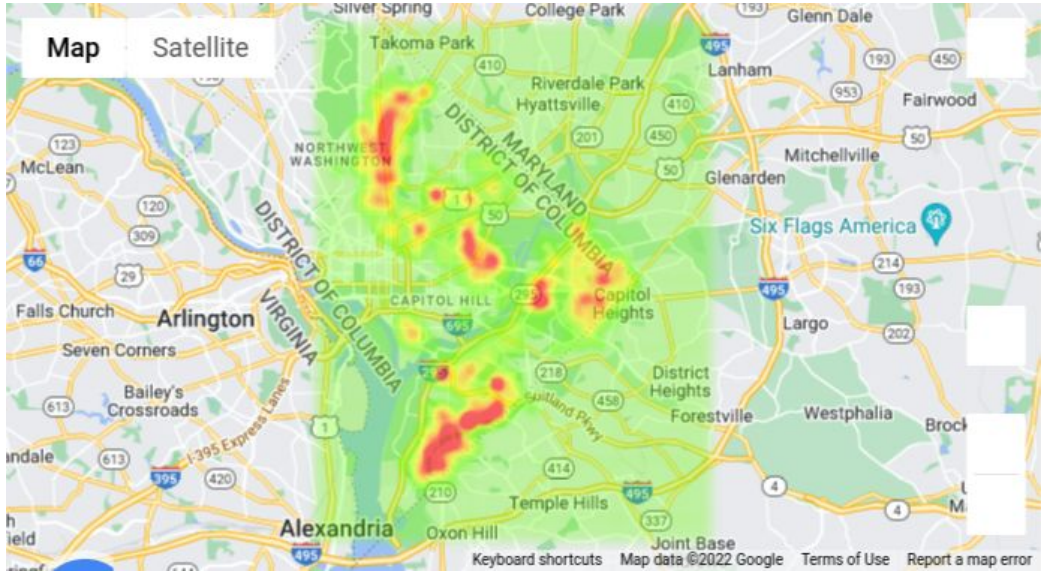

Figure 1. Example of the interactive maps developed using GMaps plugins.

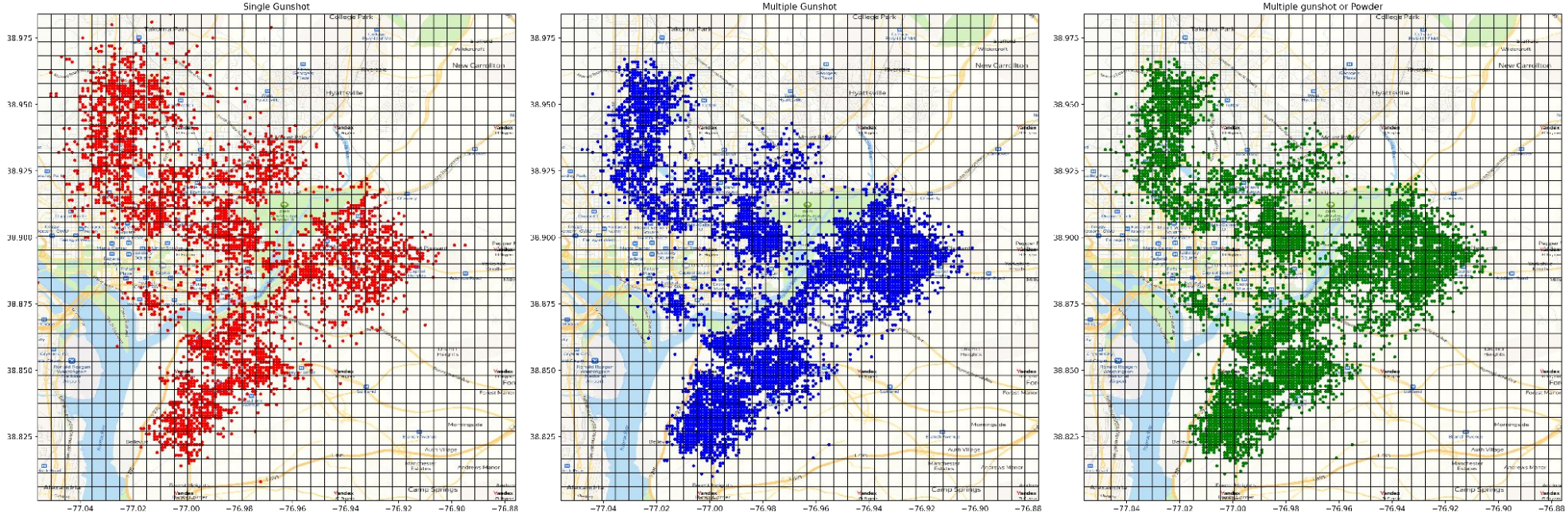

Figure 2. Example of the grid spacing inside DC and the corresponding types of gunshots.

IV. MODEL ALGORITHMS

We utilize a class of recurrent neural networks known as Long Short Term Memory (LSTM). LSTM is designed to deal with temporal data, such as those used for modeling time-series domains. LSTM is capable of learning from long term dependencies from the dataset. The architecture design of LSTM consists of three different gates, which consists of *input*, *output*, and *forget* gate. Each of the gates has their purpose in modeling the dataset. The inputs gates decides whether or not to feed the inputs data, output gates decides if it will produces a output signal of the current state, and forgot gate decides if it will forget the temporal history. To incorporate spatial dependencies into the LSTM, we utilize the work done by Liu et al [8, 6] to predict users' future movements.

V. MODEL DEVELOPMENT

We employed a combination of *Conv3D* + *ConvLSTM* + *MaxPooling* layers as our model architecture [11]. The *ConvLSTM* is a modification to the LSTM layer employed for spatio-temporal prediction. It employs convolutional operations to adjust both the input-to-date and state-state transitions. the ConvLSTM can be described as follow:

$$it = \sigma(Wxi * Xt + Whi * Ht-1 + Wci \odot Ct-1 + bi) \quad [1]$$

$$ft = \sigma(Wx\,f * Xt + Whf * Ht-1 + Wc\,f \odot Ct-1 + bf) \quad [2]$$

$$Ct = f(t) \odot ct-1 + it \odot \tanh(Wxc * Xt + Whc * Ht-1 + bc) \quad [3]$$

$$ot = \sigma(Wxo * Xt + Who * Ht-1 + Wco \odot Ct + bo) \quad [4]$$

$$Ht = ot \odot \tanh(Ct) \quad [5]$$

where red portions indicate the additional convolution operations from spatial parameters. The final model consists of multiple *ConvLSTM* layers with a *BatchNormalization* and *Maxpooling3D* layers in between. The purpose of the pooling layer is to aggregate the layer size such that the final layers will only consist of a one single layers of *Convolutional3D* layer which results in outputs of a single frame to serve as prediction. The original model architecture results in a singular location where potential events will be predicted to occur given our model. However, we instead wish to generate a heatmap where events would occur. As an result, we modified our models architectures from the Holm et al [3]. Compared to that model, we applied the following modifications to the model architectures:

- Two additional layers of *Conv3D*
- *BatchNormalization* between ConvLSTM layers
- *Bidirectional wrapper* for each of the ConvL STM layers

There are reasons that we modified the model as mentioned above. The addition of the Conv3D layer in the beginning allows more spatial features to be extracted from spatial data. Secondly, *Batch Normalizations* scaled the outputs to have mean 0 and variance of 1, which allows the neural network to ignore the in-variance and poor parameter initialization. The last modification involves the bidirectional wrapper. The reason is that the wrapper can improve the model for problems where all timesteps of the input sequence are available. Bidirectional LSTMs will train two instead of one LSTMs on the input sequence. The first on the input sequence as-is and the second on a reversed copy of the input sequence.

VI. ASSUMPTIONS AND VALIDATION

Since we are interested in daily forecasting of crime in the area. We employed a training/testing/validation ratio of around 70%/15%15% (5/1/1 years), respectively. Since our predictions come from time-series predictions, we utilize a walk forward validations so that the training period (year 1-5) is used to validate the results. In a walk forward validation process, a number of periods is used as the training starting points. However, time-series predictions tend to become less accurate as they drift off. Therefore, as new data becomes available on a daily/weekly/hourly basis, the model is retrained to allow for accurate predictions. An example of walk forward validation is shown below in fig 4. The amount of time to be included in the training set is still to be determined and requires additional research in the future.

A. Classifications Metrics

The standard classification metrics involves accuracy metrics which follows the equation:

$$Acc = (TP + TN)/Total \quad [6]$$

where *Acc* is the accuracy, *TP* is the number of truth positives, *TN* is the number of truth negative, and *Total* is the total number of samples. However, this is not suitable for evaluating the effectiveness of our data as it is extremely imbalanced (i.e. meaning that the majority of the events is 0 (no shot) compared to 1 (one events)). As a result, we assigned a weights bias in the model so that the focus is not only on having the entire area where no shot occurs. The weight bias is set with a 100 to 1 with 100 being no shots and 1 being shots.

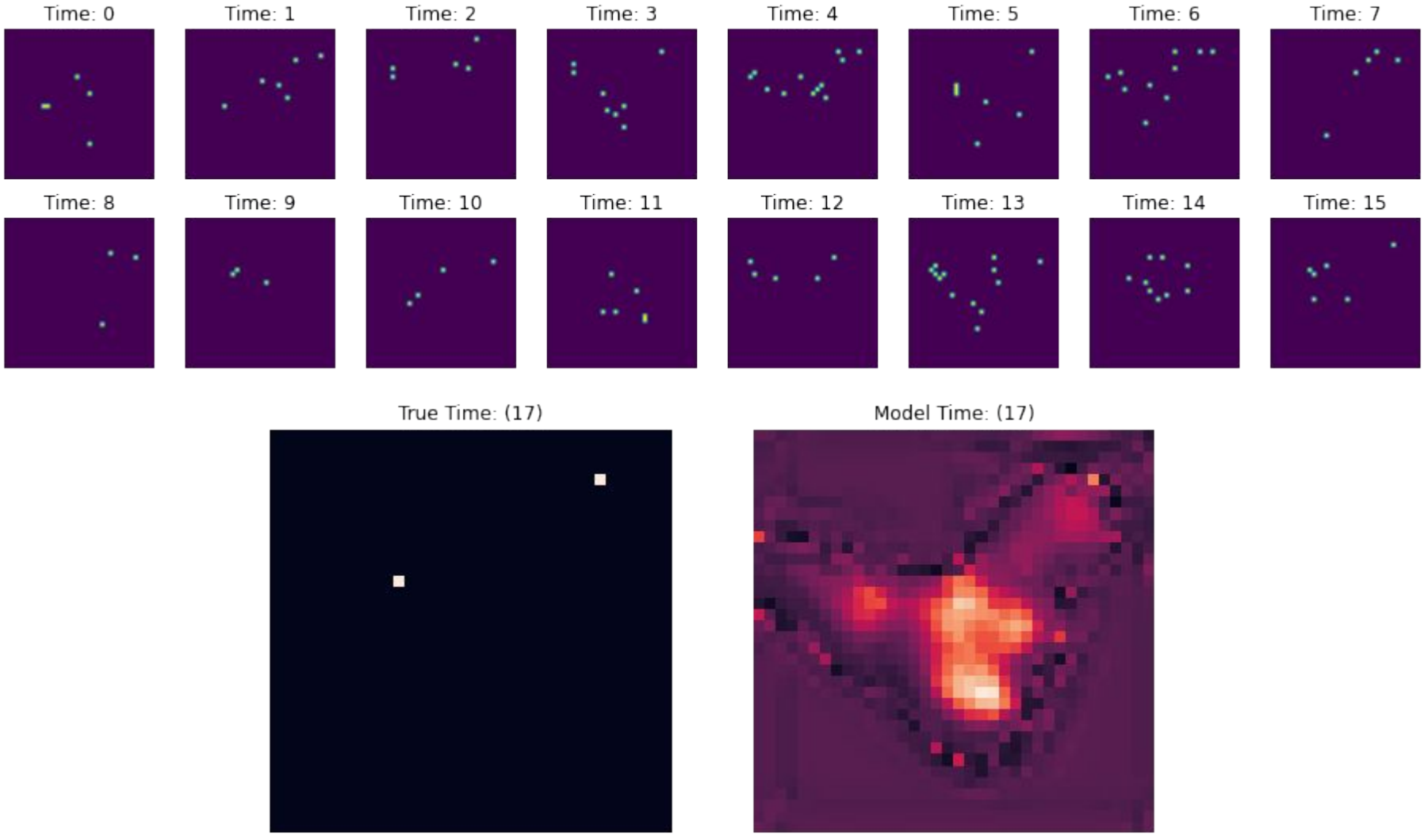

Figure 3. Example of predictions from the LSTM models.

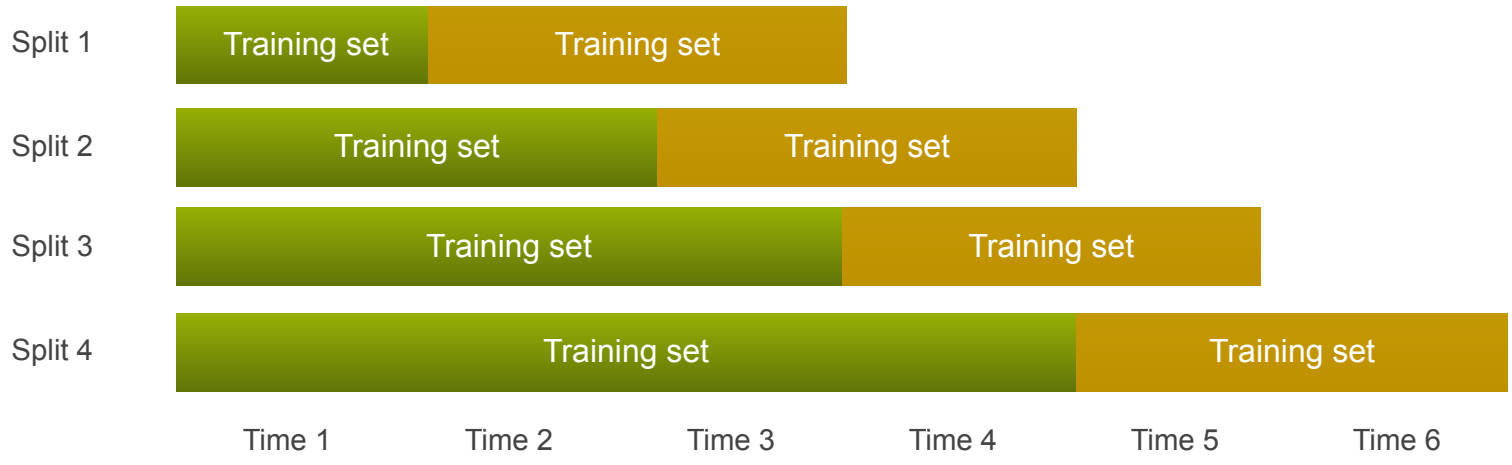

Figure 4. Example of walk forward validations.

VII. MODEL TUNING AND PERFORMANCE

We utilize the *ADAM* [5] as the optimizer and *binary crossentropy* as the loss function. The models were run for a total of 100 epochs and batch size of 20. To prevent overfitting, early termination was performed which monitors the validation loss. Learning rate is also automatically adjusted based on the changes due to plateau. Cross validation was employed with four different splits with each split containing a different amount of training set. Due to the amount of training required, the parameters optimizations were performed using Amazon SageMaker SDK [9, 10], which utilizes a Bayesian related search technique to discover the optimal hyper-parameters set. Sagemaker allows for various selection when dealing with hyper parameters optimizations, such as *ContinousParameters*, *IntegerParameters*, and *CategoricalParameters*. Afterward, we utilize regex to extract the metrics from AWS Cloudwatch and compute the optimal hyper-parameters.

VIII. MODEL RESULT

The overall accuracy of the model is determined to be around 72.2% given the input maps. The metrics accuracy however only assumes around 10:1 weights bias.

IX. DATA SCIENTISTS OWN CONCLUSIONS

There are many improvements that still need to be made, but the goal of this report is to demonstrate the first working product resulting from this working model for helping law enforcement. There are still many areas of improvements mentioned below in the next sections such as more labels and classes to the dataset, improvement to the architectures and many others.

X. CONCLUSION AND FUTURE CONSIDERATIONS

Even though LSTM is very powerful in predictions for time-series problems, it can be rather difficult in training since the structure of RNN contains a large number of variables. In the future we might utilize transformer based neural networks which are currently state of the art in NLP. Transformer performs much faster than LSTM since the information is digested all at once. Moreover, it's rather difficult to perform transfer learning in LSTM networks, but it has shown since the CNN weights We only utilized features that are included in the ShotSpotter Alerts dataset. In the future, including more relevant features or contributing factors might increase the model accuracy. Such factors or features could be access level to guns, income inequality level, concentrated poverty rate, etc. Additionally, the next version of the model involves significant reduction in the amount of spatial and temporal periods such that we can rank the street with the highest probability.

References

[1] Francois Chollet et al. *Keras*. 2015. URL: https://github.com/fchollet/keras.

[2] Dingxiong Deng et al. "Latent space model for road networks to predict time-varying traffic". In: *Proceedings of the 22nd ACM SIGKDD international conference on knowl edge discovery and data mining*. 2016, pp. 1525–1534.

[3] Noah Holm and Emil Plynning. *Spatio temporal prediction of residential burglaries using convolutional LSTM neural networks*. 2018.

[4] Plotly Technologies Inc. *Collaborative data science*. 2015. URL: https://plot.ly.

[5] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[6] Dejiang Kong and Fei Wu. "HST-LSTM: A hierarchical spatial-temporal long-short term memory network for location prediction." In: *IJCAI*. Vol. 18. 7. 2018, pp. 2341–2347.

[7] Lingbo Liu et al. "Contextualized spatial– temporal network for taxi origin-destination demand prediction". In: *IEEE Transactions on Intelligent Transportation Systems* 20.10 (2019), pp. 3875–3887.

[8] Qiang Liu et al. "Predicting the next location: A recurrent model with spatial and temporal contexts". In: *Thirtieth AAAI conference on artificial intelligence*. 2016.

[9] Valerio Perrone et al. "Amazon Sage Maker automatic model tuning: Scalable black-box optimization". In: *arXiv preprint arXiv:2012.08489* (2020).

[10] Valerio Perrone et al. "Amazon sagemaker automatic model tuning: Scalable gradient free optimization". In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge*

[11] Mader K Scott, What is ConvLSTM? https://www.kaggle.com/code/kmader/what-is-convlstm/notebook. 2018.

[12] Jumbo Zhang et al. "DNN-based prediction model for spatio-temporal data". In: Proceeding of the 24th ACM SIGSPATIAL international conference on advances in geographic information systems. 2016, pp. 1-4.