Highlights

Modeling the Need for an Ambulance based on Automated Crash Reports from iPhones

J. Bradford Burkman,Miao Jin,Xiaoduan Sun

- Supports transferability and benchmarking of different approaches on a public large-scale dataset. The associated GitHub page provides the code and detailed notes used to perform the analysis on the Crash Report Sampling System.
- Novel Application: Machine Learning Classification Models for Dispatching Ambulances based on Automated Crash Reports
- Perennial Machine Learning Challenge: Imbalanced Datasets, more challenging because we use two similar datasets with different imbalances.
-

Modeling the Need for an Ambulance based on Automated Crash Reports from iPhones

J. Bradford Burkman[a,b,*], Miao Jin[a] and Xiaoduan Sun[c]

[a] *School of Computing and Informatics, University of Louisiana at Lafayette, USA*
[b] *Louisiana School for Math, Science, and the Arts, USA*
[c] *Department of Civil Engineering, University of Louisiana at Lafayette, USA*

ARTICLE INFO

Keywords:
Automated crash notification
Ambulance dispatch
Emergency medical services
Machine learning

ABSTRACT

Put abstract here.

1. Introduction

1.1. Motivation

A Google Pixel phone can detect the deceleration profile of a car crash and, if you have enabled the settings in the Personal Safety app, will, if you do not respond in 60 seconds, automatically call the police, reporting your location. Apple announced in November 2021 that it was planning to do something similar.

The crash victims who would most obviously benefit from such technology are those in crashes with no witnesses to call police ("unnoticed run-off roadway"), who survived the crash, and might have lived if help had arrived promptly, but died from their injuries. Such crashes, though, are very rare, about seventy-seven fatalities annually in the US in 2010-2018, (Spicer, Bahouth, Vahabaghaie and Drayer, 2021) of the about 35,000 crash fatalities per year in the same time period. (NHTSA, 1975-2020)

A much larger group who could benefit from are those injuries are serious and need prompt medical attention. Dispatching an ambulance automatically, rather than waiting for an eyewitness to call for one, would cut at least several minutes off of the ambulance response time. In a 1996 study on 1990 data for US urban interstates, freeways, and expressways (Evanco, 1996), the average accident notification time was 5.2 minutes, and the additional time to EMS (emergency medical services) arrival was 6.2 minutes. Evanco estimated that reducing the notification time from 5.2 minutes to 2 minutes would cut fatalities by 15.9%. Even those who might not die may recover more fully and quickly with prompt medical attention, so dispatching an ambulance promptly when one is needed would be beneficial.

On the other hand, we do not want to send an ambulance to every accident scene, because only a small proportion of crashes have severe injury; most are property damage only (PDO) crashes. Ambulances and their crews are expensive and in finite supply.

Given the information available to the police from a phone's automated crash notification, can we build a model that will recommend (or determine) whether to send an ambulance immediately?

1.2. Datasets

For this study we used two datasets, the US Department of Transportation's National Highway Transportation Safety Administration (NHTSA)'s Crash Report Sampling System (CRSS) 2016-2020 and a census of crash reports in Louisiana 2014-2018. (NHTSA, 2016-2020) Both datasets record about five hundred thousand crashes, with details on the crash and on each vehicle and each person involved. The two datasets are similarly structured.

bradburkman@gmail.com (J.B. Burkman)
www.lsmsa.edu (J.B. Burkman)
ORCID(s):

The major difference between the two datasets is that, while the Louisiana is a census (all police crash reports), the CRSS is sample that intentionally over-represents more serious crashes. [*Give percentages of crash persons transported to hospital in each dataset.*]

We will look at the data at the level of each person involved in a crash ("crash person"), as each automated crash notification corresponds to a phone, which corresponds to a person.

1.3. Difficulty in Solving Problem

Such a model will not be perfect, with some false negatives (not sending an ambulance when one is needed) and many false positives (sending an ambulance when one is not needed). We will show that the quality of the model depends largely on what information is available. Some information (location, time of day, day of week, weather) either comes with the automated report or is easy to get. Other information (age and sex of phone's primary user, vehicle likely to be driven by that person) may be very helpful in predicting injury, but getting that information would require instantaneous communication between private and public databases. Being able to interpret the location, (*e.g.* Is that precise location inside an intersection of two roads with high speed limits?) in real time would require planning and preparation.

The problem is both political and ethical as well as technical. How many false positives will we tolerate to have one fewer false negative? We will show that, given such a marginal tolerance p, we can incorporate that tradeoff into the model, but each locality will have to decide that for itself. Implementing such a system would require budgets, cooperation, and possibly legislation, but knowing which data is most useful can help set priorities.

1.4. Machine Learning Challenges

We deal with several machine learning challenges in our study, and their solutions are often as much art as science.

Two Datasets The Louisiana dataset is a census, and the CRSS a sample intentionally weighted towards more serious crashes, so comparing the results will require interpretation and justification.

Data Imbalance Only a small proportion of crashes require immediate medical attention. Reasonable people can disagree on the number, but say it's 2%. If we built a model that classified all crashes as "Ambulance Not Needed," the model would have 98% accuracy, which would be excellent in some other applications, but not here. The toolkit for building models on imbalanced data is well established, but many of the tools only work for continuous data (our data is all categorical), and their use is as much art as science.

Feature Selection We need to pick the features most relevant to crash severity; too many less-relevant features will muddle the model building. CRSS has both "Make" and "Body Type." Do these two features give enough different information that we should use both? If not, which is more useful?

Feature Engineering - Single Feature CRSS has ninety-seven different categories of vehicle body type, and our model building would work better if we could condense those to fewer than ten categories. Is a compact pickup truck more like a car or a standard pickup? We look at the likelihood that a crash vehicle's occupants require transportation to a hospital when making such classification decisions, and classify the compact pickup truck with the cars.

Feature Engineering - Combining Features In both data sets, we have "Day of Week" and "Hour." We would like to take from each to make "Rush Hour," if it has a different hospitalization profile. When does it start and end? Is morning rush hour different from evening? Does it start earlier on Fridays?

Metrics Machine learning algorithms work iteratively by evaluating a model, perturbing the model, evaluating the new model, and deciding whether the new model is better; repeat. What do we mean by "better," and how do we measure it?

Hyperparameter Tuning The ML algorithms we will use have some user-set parameters to optimize the model, and they can only be set by trial and error. As we use two datasets, and as we add data to our model, should we use the same hyperparameters, or can they change?

Missing Data As with all real data, many samples (records) have missing values. The Louisiana data is raw, with many values listed as "unknown." For many features (fields, columns) of the CRSS, the authors have created another feature with the missing values imputed. Trusting that those authors understand the data better than we do and may have access to redacted data, we use the imputed features. For the Louisiana data, we will have to impute, delete, or ignore missing data; again, as much art as science.

1.5. Research Plan

1. On both raw data sets, do cleanup, feature selection, and feature engineering. To the extent possible, make the two engineered data sets the same.
2. Starting with the easiest-to-obtain data (general location, time/day, weather), and interatively adding more data (persons, vehicles, specific location), build and evaluate a model that predicts whether an ambulance is needed.
3. Combine results from the two datasets.
4. Interpret and discuss how the model improves as more data becomes available.

2. Literature Review

2.1. Apps to Detect Crashes

The idea of using the accelerometer in a cell phone to detect a crash and notify the police goes back to at least 2011 with *WreckWatch*, an app prototype by White, Thompson, Turner, Dougherty and Schmidt (2011). A few years later, Aloul, Zualkernan, Abu-Salma, Al-Ali and Al-Merri (2014) proposed an app that would detect not only a crash but its severity, and send the phone owner's medical information. In September 2019, Google hinted that it would have crash detection in its Pixel phones soon (Rahman, 2019), and it is now available (Google). In November 2021, the Wall Street Journal announced that Apple was thinking of introducing such an app in its iPhones and Apple watches in 2022 Winkler (2021), but as of this writing, such an app had not come out yet. The App Store does have some unverified third-party apps, like those from Sosmart SAP and BlinkApp LLC.

2.2. Imbalanced Data

In building a model to predict, based on records of roadside inspections, traffic violations, and previous crashes, future crashes for trucks, Lack, Berkow and Gao (2021) described the imbalanced data problem well.

> Initial models "correctly" classified no-crash versus crash instances 99% of the time, but almost never correctly predicting a crash—a major failure in achieving the goal of this analysis and a common issue in unbalanced datasets.

Our work in this paper uses the most straightforward of machine learning (ML) algorithms, the binary classifier. We want to answer, "Should we send an ambulance to this crash," and are using historical data answering for each crash, "Did we use an ambulance?" We want to build a model that will look at the data we have for a crash and return a prediction. To build it, we will separate our data into a training set and a test set; use the training set to build the model, then evaluate the model on the (unseen during model building) test set.

The model will not be perfect. We want to send ambulances to all of the crashes that need one (True Positive), and not send ambulances to crashes that don't (True Negative). Sending an ambulance when one is not needed (False Positive) is unnecessarily expensive, but if we don't send an ambulance when one is needed (False Negative), someone might die unnecessarily.

We can visually organize the success and failures of our binary classification model in a *confusion matrix* (also called *error matrix*, or *contingency table*).

		Prediction	
		N	P
Actual	N	TN	FP
	P	FN	TP

Most machine learning algorithms are designed to maximize *accuracy*, the proportion of classifications that were successful.

$$\text{Accuracy} = \frac{\text{TN} + \text{TP}}{\text{TN} + \text{FP} + \text{FN} + \text{TP}}$$

In many applications, the accuracy is straightforward and useful, but not if the data are imbalanced. In our case, in all of the reported crashes in Louisiana in 2014-2018, there were 645,748 people involved in the crashes, and 55,164 used an ambulance (`OCC_MED_TRANS_CD = A`), 8.5% of the total.

If we built a model that predicted that none of the crashes required an ambulance, our model would have 91.5% accuracy.

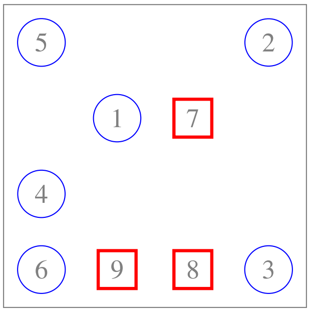
		Prediction	
		N	P
Actual	N	590,584	0
	P	55,164	0

$$\text{Accuracy} = \frac{590{,}584 + 0}{590{,}584 + 0 + 55{,}164 + 0} \approx 0.915$$

In many applications, 91% accuracy would be good. Why is it different in our context? Because the costs of false positives and false negatives are different. The cost of a false positive (sending an ambulance when one is not necessary) is measured in money, while the cost of a false negative (not sending an ambulance promptly when one is necessary) is measured in lives. We are willing to trade off sending p number of unnecessary ambulances if it means that we will send one more necessary ambulance. [The value of p is a question for ethicists and politicians that we will discuss later.]

2.3. Data-Level Techniques for Imbalanced Data

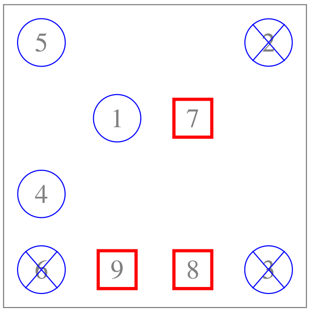
Consider this two-dimensional training dataset, which we will use to illustrate resampling techniques for imbalanced datasets. In real problems, of course, the dataset could have tens, hundreds, or thousands of dimensions. The six blue circles represent samples (elements) of the majority negative class ("no ambulance"), and the three red squares represent the minority positive class ("ambulance").



Many algorithms, and variations thereon, have been proposed to balance the two classes before applying a machine learning algorithm to build a model to classify new samples as positive or negative. WARNING: Vast oversimplification ahead. Our goal here is to give the general idea of each method.

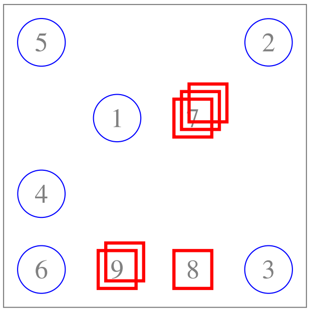
2.3.1. Random Undersampling

Random undersampling balances the two classes by randomly deleting elements of the majority class until the two are balanced. The major drawback of this method is that you throw away information about the majority class. If the majority class is many more times the size of the minority, you lose almost all of the data.



2.3.2. Random Oversampling

Random oversampling creates duplicates of minority class samples until the sets are balanced. This method has a similar effect to using class weights, introduced below.

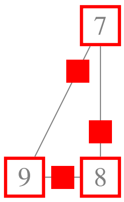


2.3.3. Synthetic Minority Sampling TEchnique (SMOTE)

SMOTE (Chawla, Bowyer, Hall and Kegelmeyer, 2002) is one of the most popular oversampling methods for balancing a dataset with continuous numerical data. It creates new synthetic minority samples "between" original minority samples, not necessarily at the midpoint by choosing a number in $(0, 1)$, multiplying the difference (in each dimension) from point A to B by that constant, and adding it to A.

One challenge with SMOTE is that it is only useful for datasets with continuous numerical data, and our data is almost all categorical. What is between "car" and "school bus," or between "parking lot" and "highway"? We will not be able to use SMOTE or similar techniques for our work.

SMOTE ignores the majority class, and we have done that in the illustration.

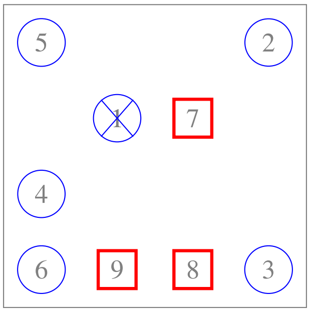


2.3.4. Tomek's Links

Tomek (1976) proposed a method of undersampling that assumes that the majority and minority classes should (at least locally) be clustered. If an sample A of the majority class and a sample B of the minority class are each other's nearest neighbors, then one of them is not clustered with its own class. Since we are trying to undersample the majority class, assume that the element of the majority class is noise (or an error, or just not useful), and delete it.

In the context of modeling crash severity from police reports, why would sample #1 not need an ambulance when its characteristics are so close to those of #7 and not near most of the other crashes without serious injury? The reason could be errors in the records, or luck/providence/fate. It could also be that the difference between property damage only and serious injury is influenced by thousands of variables we cannot measure or know, all of the physics of crash forces acting on the bones and structures of the human body. The best we can say is that the outcome in #1 cannot be predicted by the information that we have, so that sample will not help in constructing a model based on the available data; therefore, we can reasonably delete it from the training set.

Tomek's Links can also be run iteratively. Sample #7 had #1 as its nearest neighbor (both majority class), but once #1 is deleted, then #2 and #7 are each other's nearest neighbors of different classes, and we can delete #2.



2.3.5. *Bagging*

"Bagging" is short for Bootstrapped Aggregating, a variation on random undersampling (Breiman, 1996). In general, bagging takes many random subsets (with replacement) of the samples, run the classifier on each subset, then aggregate the results. In imbalanced data applications, each subset of the samples is all of the n minority samples and n randomly chosen majority samples.

In our example, bagging would make a subset of the data with the three minority-class samples (#7, 8, and 9), and three randomly chosen from the majority-class samples, run the classifier; repeat some number of times. Use an ensemble classifier to merge the results.

2.3.6. *Oversampling Image Data*

Extracting knowledge from a database of tabular numerical or categorical data is difficult, but a database of images is a challenge of a different magnitude. An imbalanced labeled image dataset for crash prediction modeling might be a thousand images taken ten seconds before a crash and a million images taken ten seconds before ... nothing happened. Deep neural networks (DNN) and (deep) convolutional neural networks (DCNN and CNN) are common methods for image data. Goodfellow, Pouget-Abadie, Mirza, Xu, Warde-Farley, Ozair, Courville and Bengio (2014) introduced Generative Adversarial Networks, which can be used to generate synthetic samples to balance the dataset. Given the power of the tools for image recognition, many researchers make non-image data look (to the computer) like images to take advantage of the tools.

2.4. Modifying the Loss Function

Machine learning algorithms generally work iteratively by picking a starting point for the constants in the model (often a random guess), measuring the error, making a small perturbation in the model constants, measuring the new error in the candidate model, and comparing the two. If the new error is less, use the candidate model; if not, go back to the old one. Repeat.

A common way to measure the error in binary classification is log loss (binary cross-entropy loss, logistic loss). For each sample in the training set we have the answer to the question (the *label*), 0 for no ambulance, 1 for ambulance, and the candidate model gives a probability $p \in (0, 1)$ that this sample will need an ambulance. The log loss is the sum over the samples of the log of the error. If the sample is in the majority class, the true value is 0, and if the model gives a value of p, then $\log(1-p)$ gives $\log(1) = 0$ if the model is perfectly correct and $\log(0) = -\infty$ if the model is perfectly wrong on this sample. Similarly, for a sample in the minority class, $\log(p)$ gives $\log(1) = 0$ if the model correctly classifies the sample. If y is the label, then the log loss for each sample is (Pedregosa, Varoquaux, Gramfort, Michel, Thirion, Grisel, Blondel, Prettenhofer, Weiss, Dubourg, Vanderplas, Passos, Cournapeau, Brucher, Perrot and Duchesnay, 2011)

$$L(y, p) = -\left(y \log(p) + (1-y)\log(1-p)\right)$$

Note that if $y = 0$, then the first term is 0 for any value of p, so only the second term (majority class) is relevant; correspondingly, if $y = 1$, then only the first term is relevant, so a clearer expression might be

$$L(y, p) = -\Big(\big[\text{Loss if minority class } (y = 1)\big] + \big[\text{Loss if majority class } (y = 0)\big]\Big)$$

Since the logs of $p \in (0,1)$ will be negative, the negative in front makes the loss positive, and the iterations of the algorithm will seek to minimize it.

2.4.1. *Class Weights (Sample Weights, Cost Sensitive Analysis)*

Class weights change the error metric, giving more weight to misclassification of minority class samples (King and Zeng, 2001). Giving double weight to misclassified minority class samples would have the same effect as duplicating all of the negative class samples in oversampling. To achieve balance in the contribution of the two classes to the loss function,

$$\text{Let } r = \frac{\text{Total number of samples}}{\text{Number of minority samples}} \qquad \text{Let } \alpha = \frac{r}{r+1} \qquad 1-\alpha = \frac{1}{r+1}$$

$$L(y,p,\alpha) = -\left(\alpha y \log(p) + (1-\alpha)(1-y)\log(1-p)\right)$$

2.4.2. *Focal Loss*

This method is recent, but has appeared in the crash analysis (see Yu, Wang, Zou and Wang (2020) referenced below). Focal loss (Lin, Goyal, Girshick, He and Dollár, 2017) adds another factor to the loss function that gives more weight to samples that are badly misclassified, and less weight to samples that are slightly misclassified.

$$L(y,p,\alpha,\gamma) = -\left(\alpha y \log(p)(1-p)^{\gamma} + (1-\alpha)(1-y)\log(1-p)\,p^{\gamma}\right)$$

The paper by Lin et al. tested values of $\gamma in[0.0, 5.0]$, and found that $\gamma = 2.0$ gives good balance, but the best value depends on the dataset and the goals. Yu et al's paper using focal loss for real-time crash prediction allowed the γ for minority and majority classes to have different values.

$$L(y,p,\alpha,\gamma_1,\gamma_2) = -\left(\alpha y \log(p)(1-p)^{\gamma_1} + (1-\alpha)(1-y)\log(1-p)\,p^{\gamma_2}\right)$$

2.4.3. *Comparison of Loss Functions*

The table below compares the three loss functions. For α, we will assume that the minority class is 10% of the dataset, so $r = 10 \rightarrow \alpha = r/(r+1) = 10/11 \approx 0.9090$. For focal loss, we will use $\gamma = 2$.

Machine learning algorithms use the loss function when comparing two candidate models, only asking which one is less, with no concern for the actual magnitude. For this reason, the choice of base for the logarithm is inconsequential (we arbitrarily choose base 10 for the chart below); also, that the raw focal loss values are each less than the raw class weights, which are each less than the raw log loss, is not relevant, so we have included normalized values for comparing the three loss functions.

Class	y	p	Raw			Normalized		
			Log Loss	Class Weights	Focal Loss	Log Loss	Class Weights	Focal Loss
Minority	1	0.9	0.04576	0.04160	0.00042	0.04560	0.08292	0.00464
	1	0.7	0.15490	0.14082	0.01267	0.15438	0.28069	0.14136
	1	0.5	0.30103	0.27366	0.06842	0.30002	0.54548	0.76309
Majority	0	0.5	0.30103	0.02737	0.00684	0.30002	0.05455	0.07631
	0	0.3	0.15490	0.01408	0.00127	0.15438	0.02807	0.01414
	0	0.1	0.04576	0.00416	0.00004	0.04560	0.00829	0.00046

Note that, with focal loss, most of the total loss comes from the one badly misclassified minority sample, so to minimize the loss, the algorithm needs to do a better job classifying that sample.

2.5. Other Algorithm-Level Methods

2.5.1. *Boosting*

Boosting is an iterative method that runs the classifier multiple times. At the end of each iteration, it determines which samples would be misclassified under the current model. In the next iteration, the classifier gives higher weight to the misclassified samples, improving the model on marginal cases. While boosting is not just for imbalanced data, the challenge in imbalanced data is that the minority class samples get misclassified, so boosting would help. A popular implementation is AdaBoost, introduced by Freund and Schapire (1997).

2.6. Imbalanced Data in Crash Analysis

Imbalanced data is a frequent concern in crash analysis. In crash prediction, "non-crash" samples are much more numerous than "crash" samples.

2.6.1. *Oversampling*

Parsa, Taghipour, Derrible and Mohammadian (2019) used SMOTE to balance the dataset of crashes on Chicago's Eisenhower expressway. Li, Abdel-Aty and Yuan (2020) used SMOTE to study crashes on urban arterials. Guo, Zhao, Yao, Yan, Su, Bi and Wu (2021) used SMOTE to consider risky driving behavior in crash prediction. Orsini, Gecchele, Rossi and Gastaldi (2021) used SMOTE in building a a real-time conflict prediction model. Elamrani Abou Elassad, Mousannif and Al Moatassime (2020) used SMOTE in studying crash prediction for collision avoidance systems. Morris and Yang (2021) compared three oversampling methods, random over-sampling, SMOTE, and adaptive synthetic sampling, for crash data analysis. Yahaya, Guo, Fan, Bashir, Fan, Xu and Jiang (2021a) compared three oversampling methods studying contributing factors in fatal crashes in Ghana, and the same group applied the majority weighted minority oversampling (MWMOTE) method to study multiple fatal injury crashes (Yahaya, Guo, Jiang, Bashir, Matara and Xu, 2021b).

2.6.2. *Multiple Techniques*

Schlögl, Stütz, Laaha and Melcher (2019) used SMOTE and maximum dissimilarity undersampling to balance an Australian dataset to build a model to predict crashes, while in a later paper Schögl used a balanced bagging approach (Schlögl, 2020). Chen, Shi, Wong and Yu (2020) used SMOTE and Tomek's links in studying crash potential in lane-changing behavior. Peng, Li, Wang, Gao and Yu (2020) used undersampling, SMOTE, cost-sensitive algorithms, and boosting. Li, Li, Yuan, Lu and Abdel-Aty (2021) used the proximity weighted synthetic oversampling technique (ProWSyn) method to build a traffic violation prediction model. Chen, Lu, Fu, Sze and Ding (2022a) compared Synthetic Minority Over-Sampling Technique for panel data (SMOTE-P) with Random Under-sampling of the Majority Class (RUMC) technique, Cluster-Based Under-Sampling (CBUS), and mixed resampling to identify explanatory factors that affect the crash risk of buses in Hong Kong. Chen et al. (2020) used ENN-SMOTE-Tomek Link (EST) in estimating crash risk in lane changing.

2.6.3. *Image (or Image-like) Data*

Formosa, Quddus, Ison, Abdel-Aty and Yuan (2020) used Deep Neural Network (DNN) models with a Mean Squared False Error loss function to analyze images from front-facing cameras in cars on a UK roadway to predict crashes. Lin, Li, Jing, Ran and Sun (2020) used Generative Adversarial Networks (GAN) to overcome data imbalance for their incident detection model. Islam, Abdel-Aty, Cai and Yuan (2021) found that a variational autoencoder was more useful than SMOTE, ADASYN, and GAN for generating minority samples to balance crash and non-crash events. Basso, Pezoa, Varas and Villalobos (2021) used Deep Convolutional Generative Adversarial Networks technique with random undersampling to use image and image-like data to build an accident-prediction model for a section of highway in Chile. Man, Quddus and Theofilatos (2022) used Wasserstein GAN (WGAN) and random undersampling to study the transferability of a model built on one dataset to other datasets. Cai, Abdel-Aty, Yuan, Lee and Wu (2020) compared deep convolutional generative adversarial network (DCGAN) with SMOTE and random undersampling to study the effects of proactive traffic safety management strategies such as variable speed limits and dynamic message signs.

2.6.4. *Other*

Lack et al. (2021) used bagging in predicting crashes for trucks and finding ways to improve truck safety. Haule, Ali, Alluri and Sando (2021) used boosting in studying the effects of ramp metering on traffic safety. Yu et al. (2020) implemented the new focal loss technique in real-time crash prediction. Shi, Wong, Li, Palanisamy and Chai (2019) used undersampling to analyze factors predicting risky and safe driving. Zhu and Wan (2021) used cost-sensitive semi-supervised logistic regression (CS3LR) for hit-and-run analysis.

2.7. Imbalanced Data in Other Transportation Areas

Mohammadi, He, Ghofrani, Pathak and Aref (2019) used the Adaptive Synthetic Sampling Approach (ADASYN) on a dataset of foot-by-foot track geometry and tonnage to identify the factors that predict rail defects. Shi and Zhang (2021) developed a hierarchical over-sampling bagging method based on Grey Wolf Optimizer (GWO) algorithm and Synthetic Minority Over-sampling Technique (SMOTE) to study lane changing for autonomous vehicles. The data was severely imbalanced because lane changing is rare compared with lane keeping. Khan, Ma, Chung and Wen (2021) used SMOTE and Tomek links and "average balanced recall accuracies" for flight delay prediction. Chen, Liu, Wang and Yamamoto (2022b) used bagging for ride-hailing demand prediction.

In crash severity, most reported crashes are just property damage only (PDO), and many PDO crashes aren't even reported. Other aspects of transportation also have imbalanced data, including

- Jiang, Yuen and Lee (2020) used similar data and addressed the challenges we'll have with it.
- Elamrani Abou Elassad et al. (2020) works with several imbalanced methods. Use this paper as a model.

2.8. Ambulances

- Park and Oh (2019) has a full-page table categorizing studies of ambulance location, relocation, and dispatching using different optimization methods.

3. Methods

[More details later]

Choose Features

Organize Data in Each Feature

Feature Engineering with Pairs of Features

Deal with Data Imbalance

Build the Model with Different Data Scenarios

Repeat with Louisiana Data

4. Results

5. Conclusions

6. Discussion

7. Future Work

Funding Statement

Conflict of Interest

The authors have no relevant financial or non-financial interests to disclose.

Acknowledgements

George Broussard contributed to this work in the NSF Research Experiences for Undergraduates program.

Data Availability

The CRSS data is publicly available at the link in the references. The Louisiana crash data is not publicly available.

Technical Paper

The technical paper with more detail and the code used for the CRSS data can be found at the corresponding author's GitHub page ....

8.

CRediT authorship contribution statement

J. Bradford Burkman: Conceptualization, Investigation, Writing - original draft, Visualization. **Miao Jin:** Supervision, Methodology, Writing - review and editing. **Xiaoduan Sun:** Data curation, Writing - review and editing.

References

Aloul, F., Zualkernan, I., Abu-Salma, R., Al-Ali, H., Al-Merri, M., 2014. ibump: Smartphone application to detect car accidents. 2014 International Conference on Industrial Automation, Information and Communications Technology, Industrial Automation, Information and Communications Technology (IAICT), 2014 International Conference on , 52 – 56URL: https://ezproxyprod.ucs.louisiana.edu:2443/login?url=https://search.ebscohost.com/login.aspx?direct=true&AuthType=ip,cookie,uid,url&db=edseee&AN=edseee.6922107&site=eds-live&scope=site.

Basso, F., Pezoa, R., Varas, M., Villalobos, M., 2021. A deep learning approach for real-time crash prediction using vehicle-by-vehicle data. Accident Analysis & Prevention 162, 106409. URL: https://www.sciencedirect.com/science/article/pii/S0001457521004401, doi:https://doi.org/10.1016/j.aap.2021.106409.

BlinkApp LLC, . Blink! URL: https://www.blinkapp.net.

Breiman, L., 1996. Machine Learning 24. URL: https://doi.org/10.1007/BF00058655.

Cai, Q., Abdel-Aty, M., Yuan, J., Lee, J., Wu, Y., 2020. Real-time crash prediction on expressways using deep generative models. Transportation Research Part C: Emerging Technologies 117, 102697. URL: https://www.sciencedirect.com/science/article/pii/S0968090X20306124, doi:https://doi.org/10.1016/j.trc.2020.102697.

Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W., 2002. Smote: Synthetic minority over-sampling technique. JOURNAL OF ARTIFICIAL INTELLIGENCE RESEARCH 16, 321 – 357.

Chen, T., Lu, Y., Fu, X., Sze, N., Ding, H., 2022a. A resampling approach to disaggregate analysis of bus-involved crashes using panel data with excessive zeros. Accident Analysis & Prevention 164, 106496. URL: https://www.sciencedirect.com/science/article/pii/S0001457521005273, doi:https://doi.org/10.1016/j.aap.2021.106496.

Chen, T., Shi, X., Wong, Y.D., Yu, X., 2020. Predicting lane-changing risk level based on vehicles' space-series features: A pre-emptive learning approach. Transportation Research Part C: Emerging Technologies 116, 102646. URL: https://www.sciencedirect.com/science/article/pii/S0968090X20305611, doi:https://doi.org/10.1016/j.trc.2020.102646.

Chen, Z., Liu, K., Wang, J., Yamamoto, T., 2022b. H-convlstm-based bagging learning approach for ride-hailing demand prediction considering imbalance problems and sparse uncertainty. Transportation Research Part C: Emerging Technologies 140, 103709. URL: https://www.sciencedirect.com/science/article/pii/S0968090X22001474, doi:https://doi.org/10.1016/j.trc.2022.103709.

Elamrani Abou Elassad, Z., Mousannif, H., Al Moatassime, H., 2020. A real-time crash prediction fusion framework: An imbalance-aware strategy for collision avoidance systems. Transportation Research Part C: Emerging Technologies 118, 102708. URL: https://www.sciencedirect.com/science/article/pii/S0968090X20306239, doi:https://doi.org/10.1016/j.trc.2020.102708.

Evanco, W.E., 1996. Impact Of Rapid Incident Detection On Freeway Accident Fatalities. Technical Report. Joint Program Office for Intelligent Transportation Systems. URL: https://rosap.ntl.bts.gov/view/dot/14153. 792508; PDF; Research Paper; DTFH61-95-C00040;.

Formosa, N., Quddus, M., Ison, S., Abdel-Aty, M., Yuan, J., 2020. Predicting real-time traffic conflicts using deep learning. Accident Analysis & Prevention 136, 105429. URL: https://www.sciencedirect.com/science/article/pii/S000145751930973X, doi:https://doi.org/10.1016/j.aap.2019.105429.

Freund, Y., Schapire, R.E., 1997. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55, 119–139. URL: https://www.sciencedirect.com/science/article/pii/S002200009791504X, doi:https://doi.org/10.1006/jcss.1997.1504.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets, in: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K. (Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

Google, . Get help a!er a car crash (pixel 6 pro). URL: https://partnerdash.google.com/apps/simulator/#get-help-after-a-car-crash?l=en&model=Pixel_6_Pro.

Guo, M., Zhao, X., Yao, Y., Yan, P., Su, Y., Bi, C., Wu, D., 2021. A study of freeway crash risk prediction and interpretation based on risky driving behavior and traffic flow data. Accident Analysis & Prevention 160, 106328. URL: https://www.sciencedirect.com/science/article/pii/S0001457521003596, doi:https://doi.org/10.1016/j.aap.2021.106328.

Haule, H.J., Ali, M.S., Alluri, P., Sando, T., 2021. Evaluating the effect of ramp metering on freeway safety using real-time traffic data. Accident Analysis & Prevention 157, 106181. URL: https://www.sciencedirect.com/science/article/pii/S0001457521002128, doi:https://doi.org/10.1016/j.aap.2021.106181.

Islam, Z., Abdel-Aty, M., Cai, Q., Yuan, J., 2021. Crash data augmentation using variational autoencoder. Accident Analysis & Prevention 151, 105950. URL: https://www.sciencedirect.com/science/article/pii/S000145752031770X, doi:https://doi.org/10.1016/j.aap.2020.105950.

Jiang, F., Yuen, K.K.R., Lee, E.W.M., 2020. A long short-term memory-based framework for crash detection on freeways with traffic data of different temporal resolutions. Accident Analysis & Prevention 141, 105520. URL: https://www.sciencedirect.com/science/article/pii/S0001457519317713, doi:https://doi.org/10.1016/j.aap.2020.105520.

Khan, W.A., Ma, H.L., Chung, S.H., Wen, X., 2021. Hierarchical integrated machine learning model for predicting flight departure delays and duration in series. Transportation Research Part C: Emerging Technologies 129, 103225. URL: https://www.sciencedirect.com/science/article/pii/S0968090X21002394, doi:https://doi.org/10.1016/j.trc.2021.103225.

King, G., Zeng, L., 2001. Logistic regression in rare events data. Political Analysis 9, 137 – 163. URL: https://ezproxyprod.ucs.louisiana.edu:2443/login?url=https://search.ebscohost.com/login.aspx?direct=true&AuthType=ip,cookie,uid,url&db=edsjsr&AN=edsjsr.25791637&site=eds-live&scope=site.

Lack, C.D., Berkow, K.S., Gao, Y., 2021. Insights into motor carrier crashes: A preliminary investigation of fmcsa inspection violations. Accident Analysis & Prevention 155, 106105. URL: https://www.sciencedirect.com/science/article/pii/S0001457521001366, doi:https://doi.org/10.1016/j.aap.2021.106105.

Li, P., Abdel-Aty, M., Yuan, J., 2020. Real-time crash risk prediction on arterials based on lstm-cnn. Accident Analysis & Prevention 135, 105371. URL: https://www.sciencedirect.com/science/article/pii/S0001457519311108, doi:https://doi.org/10.1016/j.aap.2019.105371.

Li, Y., Li, M., Yuan, J., Lu, J., Abdel-Aty, M., 2021. Analysis and prediction of intersection traffic violations using automated enforcement system data. Accident Analysis & Prevention 162, 106422. URL: https://www.sciencedirect.com/science/article/pii/S000145752100453X, doi:https://doi.org/10.1016/j.aap.2021.106422.

Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P., 2017. Focal loss for dense object detection, in: Proceedings of the IEEE international conference on computer vision, pp. 2980–2988.

Lin, Y., Li, L., Jing, H., Ran, B., Sun, D., 2020. Automated traffic incident detection with a smaller dataset based on generative adversarial networks. Accident Analysis & Prevention 144, 105628. URL: https://www.sciencedirect.com/science/article/pii/S0001457519314150, doi:https://doi.org/10.1016/j.aap.2020.105628.

Man, C.K., Quddus, M., Theofilatos, A., 2022. Transfer learning for spatio-temporal transferability of real-time crash prediction models. Accident Analysis & Prevention 165, 106511. URL: https://www.sciencedirect.com/science/article/pii/S000145752100542X, doi:https://doi.org/10.1016/j.aap.2021.106511.

Mohammadi, R., He, Q., Ghofrani, F., Pathak, A., Aref, A., 2019. Exploring the impact of foot-by-foot track geometry on the occurrence of rail defects. Transportation Research Part C: Emerging Technologies 102, 153–172. URL: https://www.sciencedirect.com/science/article/pii/S0968090X18314001, doi:https://doi.org/10.1016/j.trc.2019.03.004.

Morris, C., Yang, J.J., 2021. Effectiveness of resampling methods in coping with imbalanced crash data: Crash type analysis and predictive modeling. Accident Analysis & Prevention 159, 106240. URL: https://www.sciencedirect.com/science/article/pii/S0001457521002712, doi:https://doi.org/10.1016/j.aap.2021.106240.

NHTSA, 1975-2020. Fatality analysis reporting system. https://www.nhtsa.gov/research-data/fatality-analysis-reporting-system-fars.

NHTSA, 2016-2020. Crash report sampling system. https://www.nhtsa.gov/crash-data-systems/crash-report-sampling-system.

Orsini, F., Gecchele, G., Rossi, R., Gastaldi, M., 2021. A conflict-based approach for real-time road safety analysis: Comparative evaluation with crash-based models. Accident Analysis & Prevention 161, 106382. URL: https://www.sciencedirect.com/science/article/pii/S0001457521004139, doi:https://doi.org/10.1016/j.aap.2021.106382.

Park, H., Oh, C., 2019. A vehicle speed harmonization strategy for minimizing inter-vehicle crash risks. Accident Analysis & Prevention 128, 230–239. URL: https://www.sciencedirect.com/science/article/pii/S0001457519300314, doi:https://doi.org/10.1016/j.aap.2019.04.014.

Parsa, A.B., Taghipour, H., Derrible, S., Mohammadian, A.K., 2019. Real-time accident detection: Coping with imbalanced data. Accident Analysis & Prevention 129, 202–210. URL: https://www.sciencedirect.com/science/article/pii/S0001457519301642, doi:https://doi.org/10.1016/j.aap.2019.05.014.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830.

Peng, Y., Li, C., Wang, K., Gao, Z., Yu, R., 2020. Examining imbalanced classification algorithms in predicting real-time traffic crash risk. Accident Analysis & Prevention 144, 105610. URL: https://www.sciencedirect.com/science/article/pii/S0001457519306906, doi:https://doi.org/10.1016/j.aap.2020.105610.

Rahman, M., 2019. Google accidentally rolls out "personal safety" app, confirming car crash detection is coming to the pixel. URL: https://www.xda-developers.com/google-pixel-car-crash-detection/.

Schlögl, M., 2020. A multivariate analysis of environmental effects on road accident occurrence using a balanced bagging approach. Accident Analysis & Prevention 136, 105398. URL: https://www.sciencedirect.com/science/article/pii/S0001457519308516, doi:https://doi.org/10.1016/j.aap.2019.105398.

Schlögl, M., Stütz, R., Laaha, G., Melcher, M., 2019. A comparison of statistical learning methods for deriving determining factors of accident occurrence from an imbalanced high resolution dataset. Accident Analysis & Prevention 127, 134–149. URL: https://www.sciencedirect.com/science/article/pii/S0001457518307760, doi:https://doi.org/10.1016/j.aap.2019.02.008.

Shi, Q., Zhang, H., 2021. An improved learning-based lstm approach for lane change intention prediction subject to imbalanced data. Transportation Research Part C: Emerging Technologies 133, 103414. URL: https://www.sciencedirect.com/science/article/pii/

S0968090X21004083, doi:https://doi.org/10.1016/j.trc.2021.103414.

Shi, X., Wong, Y.D., Li, M.Z.F., Palanisamy, C., Chai, C., 2019. A feature learning approach based on xgboost for driving assessment and risk prediction. Accident Analysis & Prevention 129, 170–179. URL: https://www.sciencedirect.com/science/article/pii/S0001457518310820, doi:https://doi.org/10.1016/j.aap.2019.05.005.

Sosmart SAP, . SOSmart automatic car crash detection app. URL: http://www.sosmartapp.com.

Spicer, R., Bahouth, G., Vahabaghaie, A., Drayer, R., 2021. Frequency and cost of crashes, fatalities, and injuries involving disabled vehicles. Accident Analysis & Prevention 152, 105974. URL: https://www.sciencedirect.com/science/article/pii/S0001457521000051, doi:https://doi.org/10.1016/j.aap.2021.105974.

Tomek, I., 1976. Two modifications of cnn. IEEE transactions on Systems, Man and Communications, SMC 6, 769–772.

White, J., Thompson, C., Turner, H., Dougherty, B., Schmidt, D.C., 2011. Wreckwatch: Automatic traffic accident detection and notification with smartphones. MOBILE NETWORKS AND APPLICATIONS 16, 285 – 303. URL: https://ezproxyprod.ucs.louisiana.edu:2443/login?url=https://search.ebscohost.com/login.aspx?direct=true&AuthType=ip,cookie,uid,url&db=edsbl&AN=RN290973892&site=eds-live&scope=site.

Winkler, R., 2021. Apple wants iphones to detect car crashes, auto-dial 911. The Wall Street Journal Eastern Edition , 1URL: https://ezproxyprod.ucs.louisiana.edu:2443/login?url=https://search.ebscohost.com/login.aspx?direct=true&AuthType=ip,cookie,uid,url&db=edsgao&AN=edsgcl.681029711&site=eds-live&scope=site.

Yahaya, M., Guo, R., Fan, W., Bashir, K., Fan, Y., Xu, S., Jiang, X., 2021a. Bayesian networks for imbalance data to investigate the contributing factors to fatal injury crashes on the ghanaian highways. Accident Analysis & Prevention 150, 105936. URL: https://www.sciencedirect.com/science/article/pii/S0001457520317565, doi:https://doi.org/10.1016/j.aap.2020.105936.

Yahaya, M., Guo, R., Jiang, X., Bashir, K., Matara, C., Xu, S., 2021b. Ensemble-based model selection for imbalanced data to investigate the contributing factors to multiple fatality road crashes in ghana. Accident Analysis & Prevention 151, 105851. URL: https://www.sciencedirect.com/science/article/pii/S0001457520316717, doi:https://doi.org/10.1016/j.aap.2020.105851.

Yu, R., Wang, Y., Zou, Z., Wang, L., 2020. Convolutional neural networks with refined loss functions for the real-time crash risk analysis. Transportation Research Part C: Emerging Technologies 119, 102740. URL: https://www.sciencedirect.com/science/article/pii/S0968090X20306549, doi:https://doi.org/10.1016/j.trc.2020.102740.

Zhu, S., Wan, J., 2021. Cost-sensitive learning for semi-supervised hit-and-run analysis. Accident Analysis & Prevention 158, 106199. URL: https://www.sciencedirect.com/science/article/pii/S000145752100230X, doi:https://doi.org/10.1016/j.aap.2021.106199.