Data-Driven Structural Fire Risk Prediction for City Properties

Rupasree Dey and Alan Fern

School of Electrical Engineering and Computer Science
Oregon State University
{deyr,afern}@oregonstate.edu

Abstract

Fire Departments conduct inspections to prevent fires but it is unclear how to best allocate their limited inspection resources across the properties in a city. Currently, they use their intuition and experience to decide on which properties to inspect and lack a data-driven approach that could lead to a more principled use of inspection resources. The main contribution of this paper is to investigate such an approach, based on machine learning for predicting a fire risk score for properties in a city based on historical fire-incident data. These scores can then be used to help prioritize inspection resources toward higher-risk properties. We present a case study using data from a South Dakota fire department which contains information about properties in a city along with records of fire incidents. We use this data consisting of more than 72,000 properties to train a machine learning model to predict fire risk and evaluate its ability to rank the fire risk of properties in the city. We conduct and analyze experiments with variations of XGBoost, which is well-suited to the challenges of this application, including missing data and a highly-skewed class distribution. Our evaluation of the model-generated rankings, based on ranking metrics, shows that the model significantly outperforms random rankings and other natural baselines. We also analyze the feature importance computed for the models, which provides further insight into the model behavior. This model has been integrated into an interface for displaying the rankings across a city and is ready for beta testing.

Introduction

The effective management of fire inspections is crucial for ensuring public safety and mitigating potential risks (National Fire Protection Association 2021). Fire inspections play a vital role in identifying hazards, ensuring compliance with safety regulations, and preventing fire-related incidents. However, with limited resources and a large number of properties to inspect, it becomes challenging for fire departments to prioritize their inspections effectively. In this paper, we investigate machine learning as a way of using historical fire data to prioritize fire inspections in a city.

Current methods for scheduling fire inspections typically rely on manual processes and simplistic criteria, such as property age or complaint history, or other rule-based heuristics to determine inspection priority (Moshashaei and Alizadeh 2017). While these approaches serve as a starting point, they often fail to consider various factors that could significantly impact the risk of fire incidents and the potential consequences. As a result, there is a pressing need for a more robust and data-driven approach to prioritizing fire inspections. In recent years, some data-driven approaches (Singh Walia et al. 2018) have been proposed and deployed to address this, however, implementations of the approaches are not available for use or comparison. One contribution of this work is to develop and describe an approach that is both effective, simple, and easily applicable to new data.

Since fires are rare and highly unpredictable, it is most natural to view our problem as a ranking problem over properties in a city based on their predicted fire risk. We approach the ranking problem as learning the probability that a property will have a fire or not based on features of those properties. The application poses several challenges to learning those probabilities, including a highly skewed class distribution (fires are rare relative to non-fires) and missing features at both training and testing time. To address these challenges we investigated the effectiveness of XGBoost, which has native support for missing features and imbalanced class distributions. We describe a particular feature space and evaluate a number of algorithm variations for dealing with missing values and class imbalance. The result is a configuration that produces rankings that appear very promising from the perspective of better-utilizing inspection resources.

In the remainder of the paper, we first cover the background and prior related work. Next, we describe the problem setup and the dataset used in this case study. We then discuss our feature-engineering choices, the model choices, and the training methodology. Finally, we provide an evaluation of the effectiveness of our approach and discuss the implications for fire departments and public safety.

Background and Related Work

In recent years, advancements in data-driven approaches have shown promise in accurately predicting fire risk based on historical data and various relevant factors. In this section, we present an overview of the existing literature in the field of data-driven structural fire risk prediction, focusing on key contributions, limitations, and the need for further research. Note that the data used in these systems is not open to the public, which makes it difficult to compare directly.

Hillenbrand (2016) built a system for predicting the fire risk of each block in New Orleans, USA based on survey data from the American Community Survey (ACS)(Bureau 2023). FireBird (Madaio et al. 2016) presented another pioneering work in structural fire risk prediction. They combined historical fire inspection and incident data from multiple sources and trained machine learning models on the combined feature sets. However, they were just able to incorporate 8,223 properties out of more than 20,000 properties in the dataset because of either missing or erroneous values.

For incorporating time-varying data in training, (Singh Walia et al. 2018) proposed a dynamic pipeline for spatio-temporal fire risk prediction. They built a predictive model based on historical fire incident data and parcel data from Pittsburgh, Pennsylvania and also showed a post-hoc analysis of the model's performance over weeks of deployment. As one of the limitations, they mentioned the sparsity of the available data in fire inspection and violation data and as a result, they couldn't incorporate the violation data in the training. Although direct comparison is challenging because of publicly inaccessible data, their work leads to some important ideas. The ideas include identifying high-risk clusters from the model's predictions and experimenting with different levels of aggregation for analysis and visualization.

Other work on predicting fire from multi-source data (Dang et al. 2019) proposed a fire prediction system in Humberside, UK for commercial properties. Their results show that prediction is better with a training window of three or five years. CityGuard (Wang et al. 2019) is another system for analyzing spatial and temporal factors that impact fire risk. The system utilized the GRU-CRF model to learn temporal dependencies and design a spatial-temporal loss function to learn spatial dependencies at a timestamp. The experiment was conducted on four different datasets from Zhengzhou, China from 2014 to 2018. A few of the previous works (Lee, Lin, and Madaio 2018) attempted to analyze the stability of the model's performance over a certain period since deployment. The results demonstrated that the evaluation metrics and feature importance fluctuated over the deployment period.

To compare the existing works with our system, handling class imbalance comes first. As the fire incidents are rare in the dataset, dealing with class imbalance is necessary to mitigate bias in the model's performance. To balance the classes, some of the previous works used synthetic oversampling techniques like SMOTE (Chawla et al. 2002). While this is an interesting approach to class imbalance, the use of synthetic data poses the risk of distorting the distribution of actual data and/or introducing new types of data bias. Rather, in our work, we take a more straightforward approach to class imbalance that adjust the class weights in the loss function along with resampling-based approaches. Regarding missing values, we didn't impute the missing values because every imputation technique has some assumptions and that might result in a biased dataset. We utilized the 'Sparsity Aware Split-finding Mechanism' of XGBoost (Chen and Guestrin 2016) to handle missing values. We also compare to the model's performance trained on the data that doesn't have any missing values.

Problem Definition and Dataset

To prioritize fire inspections effectively, we need to capture the relative fire risk associated with different properties. In this section, we first formulate the problem as one of probability estimation. Next, we describe the datasets used as the basis of our machine-learning approach.

Problem Formulation

Let the set of all city properties be denoted as $P = \{p_1, p_2, \ldots, p_n\}$, where n is the total number of properties. Each property p_i is associated with an m-dimensional feature vector $x_i = (f_{i1}, f_{i2}, \ldots, f_{im})$, which captures both time-variant and time-invariant attributes about the property based on city and fire-inspection records (details in Section). Each property p_i is also associated with a discrete label c_i, which indicates whether the property has experienced a fire within the time range of the dataset and if so what type of fire it experienced. In this work, we are primarily interested in predicting the fire risk, rather than the particular type of fire risk. Thus, we also let y_i denote the binary target label indicating whether or not p_i experienced a fire event or not. Our goal is to learn a model to estimate the probability of fire, $\Pr(y_i|x_i)$, for any property. This probability can be used as the basis for a fire risk ranking.

Dataset Description

The dataset used in this study consists of historical fire incident data and property characteristics. Table 1 lists some of the attributes and time-range of multiple sources of raw data in a hierarchical manner. Most of the attributes in this table are self-explanatory, with the exception of code case violations. During fire inspections, if any fire-code violations are found, such as inadequate fire exits, faulty fire suppression systems, or blocked emergency exits, they would be documented as code case violations. All of the datasets have the 'address' attribute in common. We used this attribute to join the separate datasets to prepare the comprehensive dataset. Note that for readability, the table lists only some of the attributes in each of the datasets.

The dataset exhibits certain characteristics that warrant attention. Firstly, class imbalance is observed, as fire incidents are relatively rare occurrences compared to the number of properties in the dataset. In particular, only 1.3% of the properties have a positive y_i label indicating a fire incident. This class imbalance poses challenges in training accurate predictive models and requires careful consideration during model development. Additionally, there are many missing values in the dataset, as some properties may lack complete information on certain features. Missing data can impact the performance of machine learning models and therefore, deciding on techniques to deal with missing data is crucial. Table 2 shows the missing value percentage of the features that have missing values in the dataset.

Dataset	Attributes	Time-range
Address	Location, Age, FCODE, ZIPCODE, Construction Type, Occupancy	Not applicable
Codecase Violations	Type of codecases, Time of violations	2015-2021
Crimedata	Time of crime, Type of crime	2017-2021
Utility Disconnections	Time of disconnection, Count of disconnections	2010-2018
Rental Registrations	Permit Issue Date, Permit Expiration Date, Permit Holder Name	2008-2022
Fire Incident Data	Fire Code, Incident Date, Incident Type	2011-2020
Fire Inspections	Inspection Time	2015-2021
Parcel data	SQFT, Acreage, Frontfoot, Land value, Building Value	Not applicable

Table 1: Listing time-range and attributes of multiple sources of data.

Feature	Missing(%)
OCCUPANCY	20.7
Construction Type	23.3
FCODE	0.4
Age	22.1
SQFT, Acreage, Front foot, Shape Length, Shape Area	16.2
BuildValue, LandValue	16.5

Table 2: Percentage of missing values by features.

Technical Approach and Methods

Feature Engineering

The original dataset included information about many types of fires, which did not all involve structures. To limit the scope of our investigation, we first processed the data to only include structural fires, which were indicated by fire code '111'. The raw data consists of both time-variant and time-invariant features that were handled separately. The time-varying features like inspection date, code case violation date, etc. were converted into an aggregated form over the time from 2011 to 2021. The type of aggregation depended on the feature time but typically involved additive aggregation of counts. For example, the number of inspections and the number of code violations. By aggregating time-variant features over all available years, the long-term patterns and trends related to these features are captured. This can provide insights into the overall historical behavior of properties and potentially reveal cyclic patterns that might not be apparent within shorter windows. In addition, to help encode location information, we introduced a geohash feature, which was generated from the latitude and longitude of each property. In total, we used 24 features for each property, with 14 numeric and 10 categorical. Further, we convert the categorical features to a one-hot encoding, which results in a final set of 698 features.

We arrived at the feature set in two steps. Firstly, we used the features resulting from a discussion between Levrum Data Technologies and the fire chief of the South Dakota Fire Rescue Department. Secondly, we explored the existing literature on fire inspections to build the feature set. We did not do an extensive exploration of feature subsets, which could be a potential avenue for future work.

Model Training and Evaluation

We divided the 72,564 properties into a 75/25 train-test split. To deal with the class imbalance, we ensured that 25% of both the fire instances and non-fire instances were in the test set and the remaining 75% of each in the training set. This resulted in a test set with 320 properties having fire instances out of 23,947 test properties. We performed five-fold cross-validation on the training set and used the average validation score of the five splits to tune our single hyperparameter 'scaling positive weight', which adjusts the relative weight of the positive class to help address class imbalance. The best value of this hyperparameter is used to build the final model. Finally, that model's performance was evaluated and reported on the test set.

We chose XGBoost to build the predictive model due to its scalability, stability, and fast runtimes (Chen and Guestrin 2016). During the experimental setup, the hyperparameters of XGBoost were kept as default except for the 'scaling positive weight' which was set via cross validation as described above. We also ran experiments with ADABoost and Random Forest models, using their default hyperparameters except for the negative-to-positive scaling weight, which was set via cross validation. Note that since our ADABoost and Random Forest implementations do not support missing values, we removed properties that had missing values from the dataset when using those models.

Our models estimate the probability of fire given a property, which is used to rank properties. Since fire risk is a continuum rather than a binary outcome, it's crucial to assess how well the model ranks properties according to their likelihood of experiencing a fire incident. We evaluate these rankings via the Area Under the Curve (AUC), which is common metric for evaluating rankings and, thus, aligns well with our goal of prioritizing fire inspections. AUC takes into account the model's performance across different prediction thresholds and is less affected by imbalanced classes compared to metrics such as accuracy. It provides a holistic view of the model's ability to distinguish between positive (fire) and negative (non-fire) instances.

Results & Discussion

We present two sets of experiments. First, we investigate the performance of XGBoost trained on the full dataset that includes features with missing values. Second, we compare XGBoost to ADABoost and Random Forest on the reduced dataset that removes properties that have missing values. This allows for a comparison of the three models as well as information about the importance of including the missing-value features.

Full Dataset Experiments

Table 3 shows the mean validation scores in five-folds cross-validation of XGBoost models. Each of four rows corresponds to the value of 'positive scaling weight'. We see that increasing the weight of the positive class (fire incident) results in a decrease in the AUC. This trend was initially surprising, given that larger weights are intended to better equalize the class imbalance. These results suggest that XGBoost is already able to adequately deal with the class imbalance via unweighted statistics. Another explanation for the trend may be a complicated interplay between the weighting factor and the way XGBoost handles missing features, which relies on selecting values based on dataset-level statistics.

Based on these results, we trained XGBoost on all of the training data using a weight of 1. The result AUC for the test data is 0.757, which is based on 320 fire incidences in the test data.

Pos-scale-weight	Mean Validation AUC
1	0.772
50	0.742
73	0.724
100	0.715

Table 3: AUC scores of rankings with missing data.

To further investigate the utility of the learned rankings, Figure 1, plots curves for the chosen train-test split where the X-axis denotes the number of properties sorted in decreasing order of predicted rank (higher to lower fire likelihood) and the Y-axis denotes the cumulative count of actual fires incidents within the ranked properties. Intuitively, if a fire department used the rankings to dictate the order of their inspections, the X-axis corresponds to the number of inspections and the Y-axis then corresponds to how many of those inspections involved a property that was identified as having a fire incident in the data. Thus, steeper slopes indicate more efficient usage of inspection resources. Note that here we only include the first 500 properties from each ranking. We include curves for different positive scaling weights of XGBoost as well as the performance of a random ranking, which was obtained by averaging the curves of 100 random rankings.

Overall, we see that XGBoost can significantly outperform the random ranking for each parameter value. For example, using a weight of 1 for XGBoost results in 16 properties involving fire incidents among the first 100. In comparison, the random ranking includes only a single property with fire incident in the top 100.

Reduced Data Experiments

Table 4 shows the respective AUC scores of rankings generated by XGBoost, ADABoost, and Random Forest where properties with missing values have been removed.

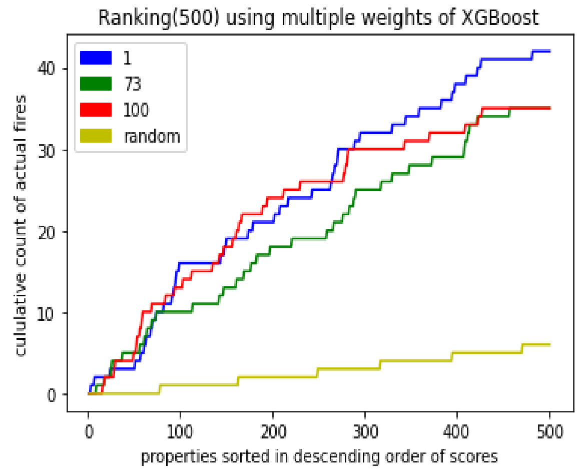

Figure 1: Comparison of random ordering with three models of XGBoost trained using weights 1, 73 and 100. The curve in blue color(one that has the highest value along y-axis) corresponds to weight '1', the curves in red and green color(the second highest value along the y-axis) correspond to weight '73' and '100' respectively. The curve in yellow(lowest value along y-axis) corresponds to random ranking.

Without Scaling		After Scaling	
Model	AUC	Model	AUC
XGBOOST	0.598	XGBOOST	0.603
ADABOOST	0.625	ADABOOST	0.656
Random Forest	0.693	Random Forest	0.699

Table 4: AUC scores of rankings on reduced dataset where features with missing values are removed. Results are shown for the algorithms run without positive scaling and with positive scaling set to perfectly equalize the positive and negative classes.

First, we see that XGBoost performs significantly worse on this reduced dataset compared to using the full dataset with missing values. This indicates that the features with missing values contained useful ranking information and that XGBoost provides an effective mechanism for dealing with the missing values. We also see that XGBoost improves in this case for the larger scaling paramter (with scaling), which is the opposite of what we observed in the experiments with missing values. This provides some evidence that there is indeed a complicated interplay between the way that XGBoost deals with missing values and the positive scaling parameter. More work is needed to understand and potentially address this negative interaction. Finally, we see that for this limited dataset both ADABoost and Random Forest are able to outperform XGBoost, with Random Forest being the top performer.

Based on the superior performance of ADABoost and

Random Forest on the reduced dataset, we explored using them on the full dataset in combination with a variety of imputation techniques using available packages. In particular, we explored the KNNImputer and LinearRegression techniques from the sklearn library. Our hypothesis for the poor performance of these techniques is that the pattern of the missing values is highly non-random and very much depends on contextual features of the properties. This complexity violates the assumptions of the imputation techniques we tried, resulting in biased data. XGBoost's approach to handling missing values apparently is able to capture the nature of the missing values more effectively since it is a highly contextual approach. However, we were unable to find a combination that is competitive with XGBoost on the full dataset. We suspect that this is due to the challenge of producing meaningful imputations for the categorical features in the dataset compared to imputing numerical feature values.

Figure 2 shows the cumulative fire curves for the three models, both with and without scaling, along with the curve for the random ranking. As expected the higher sloped curves (more efficient rankings) correspond to the larger AUCs. We see that while all methods outperform the random ranking, they do not dominate the random ranking as much as XGBoost did for the full dataset. After deleting the properties with missing values, the total number of properties reduced to 4811, among which 172 properties had fires. Before deleting the missing values, the dataset had 969 fires among 72564 properties. By deleting properties with missing values, the size of the dataset has been substantially reduced. This data reduction might have affected the representativeness of the dataset and, consequently, the performance of the machine learning model. In addition, in the reduced dataset, the class imbalance is even more pronounced. There is a smaller number of positive cases (fires) relative to the negative cases (no fires). Highly imbalanced datasets can lead to challenges in model training, where the model might have difficulty learning the minority class. Taken together these factors have a significant impact on performance and result in much worse performance in the reduced dataset experiment.

Analysis of Feature Importance

To gain some insight into the useful features for our ranking problem, we use the capability of XGBoost to rank the feature importance of a learned model. For each of the learned models (different traning sets) we obtained the feature importance rank and aggregated the ranks into an overall rank. Table 5 shows the top ranked features from most important to less important for both the full dataset and the reduced dataset. Some features such as Geohash, occupancy types, and construction types have sub-categories and hence the sub-categories are listed together under the particular features in the table.

For the full dataset, we see that the dwelling type 'Multi-Family Dwelling' plays a significant role in fire risk prediction. This kind of dwelling usually has a higher occupancy, which can strengthen the factors responsible for fire incidents. Secondly, the inclusion of particular geohash values

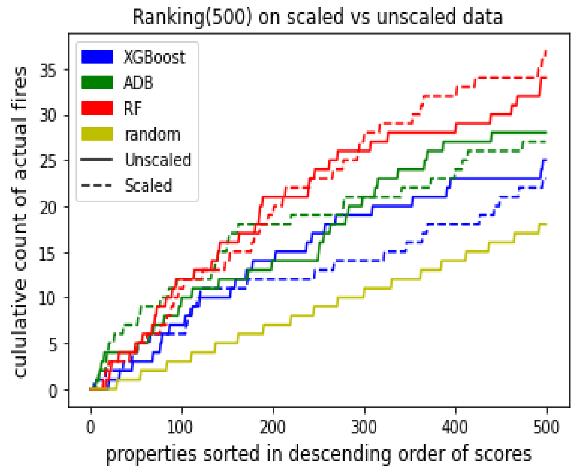

Figure 2: Comparing performance of XGBoost, ADABoost & Random Forest models on unscaled vs scaled data using cumulative fire count curve. Solid lines and dashed lines respectively denote the curve for unscaled and scaled data. Curve with highest value along y-axis corresponds to Random Forest, the second and third highest corresponds to ADAboost and XGBoost models.

in the feature importance list says that certain areas or neighborhoods might have higher fire risk factors due to population density, activities, and other factors. The ZIPCODE feature is also highly ranked and captures broader geographic patterns, compared to geohashes, which might be related to socioeconomic status, urban versus rural division, and proximity to natural hazards. We also see that the volume of crime in an area is also judged as an important factor, which may have a correlative effect to the socioeconomic status or other factors that directly impact fire incidents.

The number of prior fire inspections is also ranked as important, which indicates that buildings with more inspection history might have a lower risk of fire incidents because of better adherence to safety regulations. The utility disconnections count represents the frequency of utilities like gas or electricity disconnections in a building. A higher count suggests an increased fire risk due to potential issues with the electrical or gas systems or more general neglect, which could lead to fire incidents. Both the construction types V-N and V1-HR denote wood-framed buildings that are the most combustible out of all the types and thus have little fire resistance.

Another important feature is fire code violations and it indicates that buildings with a higher number of violations are likely to have a higher fire risk. It emphasizes the importance of compliance with fire safety regulations for mitigating fire risk. Lastly, the occupancy types 'R1' and 'B2' respectively correspond to residential and business purpose buildings. Residential buildings may have specific fire safety requirements due to the presence of households, while commercial buildings might have additional risks due to the nature of business activities. Overall, the features indicated as important agree with intuition, which can help build trust in

Full Dataset	Reduced Dataset
FCODE Multi-Family Dwelling	Age
Geohash 9zeps	Shape Length
Crime Count	SQFT
ZIPCODE	Crime Count
Inspection Count	Acreage
Utility Disconnections	ZIPCODE
Construction Type VN, V1-HR	Geohash 9zepx, 9zepw
Violations Count	Violations Count
Geohash 9zepz, 9zepx	FCODE Two-Family Dwelling
Occupancy Type R1, B2	Geohash 9zeps, 9zepr

Table 5: Feature Importance List.

the model.

For the reduced dataset, some features are common in both experiments, such as crime count, Zipcode, violations count and some geographical regions. The location identified by geohash '9zepx', '9zeps' can lead to some interesting insights into the correlation of location and fire risk. Additionally, the dwelling types, 'Two-family' or 'Multi-family' of the properties turn out to be significant predictive features of fire risk. It suggests that these factors are consistently influential in predicting fire risk regardless of the specific experiment or model. However, some features like construction type, and occupancy type carried more importance in predicting fire when using the full dataset, but they are not present in the feature list from the experiment with reduced data.

Finally, Figure 3 shows the locations corresponding to the most important geohash feature on the map. Upon inspection, we can notice that these are high density areas. We also produced an interactive visualization, depicted in Figure 4, where the orange markers show the high-risk properties and the red markers show the places with previous fires and the blue ones show other properties. The popup value on the orange marker shows the prediction score for that property. This interface could be used by fire chiefs to aid in decision support for inspection prioritization.

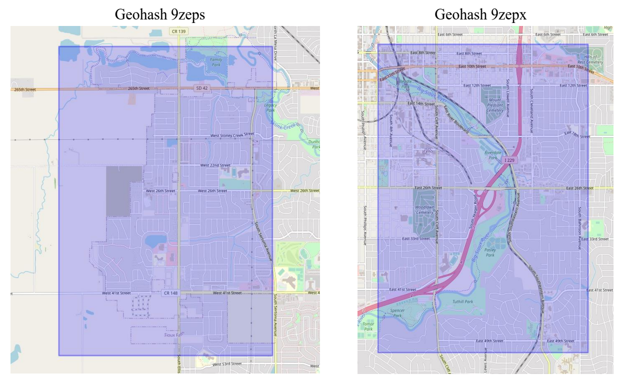

Figure 3: The image on left side shows the region on the map denoted by Geohash '9zeps' and the image on right side shows the region for Geohash '9zepx'.

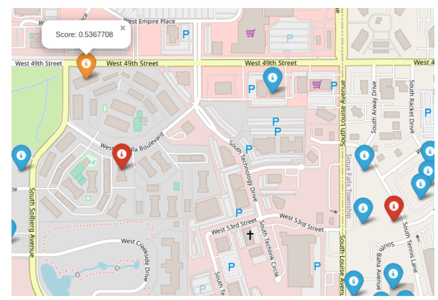

Figure 4: A snippet showing high-risk properties marked on map; yellow marker means properties with predicted high risk, red marker shows the properties with previous fire incidents, blue markers show the other properties.

Path to Deployment

The proposed system will be deployed at a Fire Rescue Department in South Dakota to help them prioritize fire inspections. We will also develop a web interface where the users can see the order of properties by their risk score and also the properties that had previous fire incidents on an interactive map. One of the deployment challenges will be to maintain the models with the latest data on fire incidents and properties, which will require retraining. There are several options for this, which need to be investigated from a cost-benefit perspective.

Summary and Future Directions

In summary, this research contributes to the advancement of fire inspection practices by providing a data-driven approach to prioritize inspections and allocate resources effectively. However, there are still opportunities for further improvement and exploration. Future directions for this research include investigating alternative machine learning models to potentially improve prediction accuracy and robustness. Additionally, exploring suitable imputation techniques for missing data in the fire incident dataset can enhance the quality of predictions. Since Random Forest worked best among the three models in the reduced data, combining appropriate imputation techniques with Random Forest could also result in higher performance. Furthermore, extending the analysis to predict other categories of fires, such as the severity or cause of fires, can provide additional insights and aid in developing targeted prevention strategies. It is also worthwhile to consider alternative evaluation metrics that capture the specific needs and priorities of fire departments. By adopting the proposed and future directions, the utility of fire risk prediction can be enhanced, ultimately leading to improved public safety and the prevention of fire incidents.

Acknowledgements

We thank Levrum Data Technologies for sharing the data, initial codebase, and overall project support.

References

Bureau, U. S. C. 2023. American Community Survey. https://www.census.gov/programs-surveys/acs. Accessed: 2023-08-13.

Chawla, N. V.; Bowyer, K. W.; Hall, L. O.; and Kegelmeyer, W. P. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16: 321–357.

Chen, T.; and Guestrin, C. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794.

Dang, T. T.; Cheng, Y.; Mann, J.; Hawick, K.; and Li, Q. 2019. Fire risk prediction using multi-source data: A case study in humberside area. In *2019 25th International Conference on Automation and Computing (ICAC)*, 1–6. IEEE.

Hillenbrand, K. 2016. Predicting fire risk: From New Orleans to a nationwide tool. *ASH CENTER, Havard Kennedy School*.

Lee, J.; Lin, Y.; and Madaio, M. 2018. A Longitudinal Evaluation of a Deployed Predictive Model of Fire Risk. In *NeurIPS Workshop on AI for Social Good*.

Madaio, M.; Chen, S.-T.; Haimson, O. L.; Zhang, W.; Cheng, X.; Hinds-Aldrich, M.; Chau, D. H.; and Dilkina, B. 2016. Firebird: Predicting fire risk and prioritizing fire inspections in Atlanta. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 185–194.

Moshashaei, P.; and Alizadeh, S. S. 2017. Fire Risk Assessment: A Systematic Review of the Methodology and Functional Areas. *Iranian Journal of Health, Safety and Environment*, 4(1): 654–669.

National Fire Protection Association. 2021. US fire problem. https://www.nfpa.org/News-and-Research/Data-research-and-tools/US-Fire-Problem. Accessed: 2023-07-14.

Singh Walia, B.; Hu, Q.; Chen, J.; Chen, F.; Lee, J.; Kuo, N.; Narang, P.; Batts, J.; Arnold, G.; and Madaio, M. 2018. A dynamic pipeline for spatio-temporal fire risk prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 764–773.

Wang, Q.; Zhang, J.; Guo, B.; Hao, Z.; Zhou, Y.; Sun, J.; Yu, Z.; and Zheng, Y. 2019. CityGuard: citywide fire risk forecasting using a machine learning approach. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(4): 1–21.