International Journal for Electronic Crime Investigation

ISSN: 2522-3429 (Print)
ISSN: 2616-6003 (Online)

DOI: https://doi.org/10.54692/ijeci.2026.1001/271

Research Article

Vol. 10 issue 1 Jan-Jul 2025

Comparative Machine Learning Models for Street Crime Prediction Using Lazy Learning Algorithms: A WEKA-Based Empirical Study

Muhammad Talha Khan [1]*, Muhammad Tayyab Waqar[2], Intakhab Alam[2], Abdul Jabbar[3], Hina Akhtar [4], Muhammad Sohaib Abdullah[5]

[1] International Collaborative Research Group, Nanchang, China

[2]Department of Computer Sciences, University of Management and Technology, Sialkot, Pakistan

[3]Department of Computer Science, university of Sialkot, Sialkot, Pakistan

[4] Department of Computer Science, NFC Institute of Engineering and Fertilizer Research, Faisalabad, Pakistan

[5]International Collaborative Research Group, Swabi, Pakistan

Corresponding Author: talhabinkhalid04@outlook.com

Received: June 15, 2026, **Accepted:** June 25, 2026; **Published:** June 28, 2026

ABSTRACT

Predicting street crime levels enables proactive policing and efficient allocation of resources. Three lazy learning algorithms, namely, k-Nearest Neighbor (IBk), K-Star and Locally Weighted Learning (LWL), are compared for community-level violent crime classification (here used as a measurable proxy for street crime) within the Low, Medium, and High categories. The Communities and Crime dataset was obtained from the UCI Machine Learning Repository and contains 1,993 cleaned community instances with 100 socio-economic and law-enforcement variables. The target attribute was discretized into three nearly balanced classes using equal-frequency binning to form the "violent crimes per 100,000 population" attribute. The implementation and evaluation of all models are performed using WEKA 3.6.14 with stratified 10-fold cross-validation and using the same 10 stratified folds for each model to allow paired statistical testing. Results demonstrate that IBk with k = 7 achieves the highest accuracy at 64.53% (95% CI: 62.47–66.58%), a weighted F-measure of 0.643, and a Kappa statistic of 0.468. This improvement in comparison with K-Star and LWL is confirmed by a corrected resampled paired t-test to be statistically significant ($p < 0.005$ for both comparisons) with a large effect size (Cohen's $d > 1.7$). Overall, LWL exhibited the poorest performance and does not correctly classify any instance from the Medium class; results similar to that of its DecisionStump base learner which is limited in expressive ability. The results in this paper indicate that neighborhood averaging instance-based learners outperform kernel-weighted and entropy-based lazy learners for this multi-class task for the prediction of crime level under the experimental settings used in this paper.

Keywords: Ransomware detection; WEKA; stacked generalisation; CIC-IDS2018; network intrusion detection; machine learning

1. INTRODUCTION

Street crime continues to be a serious issue in cities all over the world, for both law enforcement and policy makers and for residents. Predicting community-level violent crime can assist police agencies in planning patrols, allocating officers, and implementing targeted crime prevention strategies. Conventional crime analysis was primarily based on descriptive statistics and manual review of records, and did not easily account for complex, nonlinear relationships between socio-economic and law-enforcement variables. Such patterns can be captured automatically from historical records using machine learning, as described in typical data mining literature [1, 2].

Supervised classification is the most widely used machine learning approach for crime data because the number of crimes can be divided into distinct categories, e.g., Low, Medium, High. Lazy learning algorithms are among the many classification paradigms that are available. These algorithms are attractive because they defer generalization until a query instance is encountered. This property enables the natural adaptation of the lazy learner to local neighborhoods of similar communities, which can be more representative of the local characteristics of crime patterns than a single, global decision boundary.

Because publicly available datasets rarely contain information about the location of crime on a per-street segment level, this study follows common practice in the literature. Street crime is defined here as the community-level violent crime rate, which is defined as the number of murders, rapes, robberies and attacks that occurred per 100,000 people. This operational definition is applied throughout the rest of the paper and henceforth the term violent crime in the community is substituted for street crime where precision with regard to the unit of analysis is important.

There are several studies that use classification algorithms like Decision Trees, Naïve Bayes and Support Vector Machines to crime prediction tasks and a smaller number of studies that discuss some of the individual lazy learners such as k-Nearest Neighbor. A systematic, side-by-side comparison of multiple lazy learning families is still limited, however. This is the case for instance-based neighbor voting, entropy-based instance similarity and locally-weighted regression-type learning combined within one publicly available crime dataset and with the same WEKA protocol. This research gap motivated the present study.

This paper makes three primary contributions. First, it develops a reproducible pipeline based on WEKA, which transforms the UCI Communities and Crime dataset into a three-class community-level violent crime prediction task. Second, it compares the performance of three representative lazy learning algorithms, IBk with different sizes of the neighborhood, K-Star and LWL, using the same settings of stratified 10-fold cross-validation, where the parameters of the classifiers, the random seed numbers and the assignment of the folds are all reported so that the results can be reproduced. Third, it reports accuracy, precision, recall, F-measure, kappa statistic and ROC area of all models, as well as per-class confusion matrix and statistical significance tests between models,

to determine which lazy learning strategy achieves the best performance for this task.

This paper is organized as follows: In Section II, relevant prior research on machine learning for crime prediction is reviewed. The dataset, data pre-processing, algorithms and experimental setup are described in Section III. Experimental results (significance tests) are presented and discussed in Section IV. Section V concludes the paper and discusses directions for future work.

2. RELATED WORK

Initial research on crime classification did not consider specifically lazy learners, but rather compared conventional supervised algorithms. Iqbal et al. [3] used Naïve Bayes and Decision Tree classifiers on United States crime dataset with WEKA and 10-fold cross-validation, and they reported that the Decision Tree classifier outperformed the Naïve Bayes classifier in predicting the category of crime. Kim et al. [4] tested several classifiers, including k-Nearest Neighbor, for crime type classification, and found neighbor-based methods to achieve superior predictive performance compared with a purely probabilistic approach.

Wibowo and Oesman [5] made a direct comparison of k-Nearest Neighbor, Naïve Bayes and Decision Tree for crime and criminal action prediction in an Indonesian regency where they found that the type of algorithm used for the prediction of crime and criminal action is very significant in various crime categories. In the field of machine learning and computer vision-based crime forecasting and prevention, Chun et al. [6] have reported that instance-based and ensemble techniques are among the most popular ones in the literature. Khan et al. [7] created a crime prediction model using crime data from San Francisco and various classification methods, and warned that the performance of the model is affected by the discretization of crime counts into classes.

More general systematic reviews support the importance of comparative review. Mandalapu et al. [8] conducted a literature review of studies on crime prediction using machine learning and deep learning, and reported that there was a persistent demand for transparent, reproducible comparisons between different benchmarks. Kshatri et al. [9] have suggested a stacked generalization (SG) ensemble for crime prediction and demonstrated that accuracy of combined multiple base learners is better than any of them. Butt et al. [10] reviewed the methods for detecting spatio-temporal crime hotspots and pointed out that the selection of an appropriate algorithm depends on the representation of the spatial and demographic features.

More concretely, with regards to street crime specifically, Yunus and Loo [11] used a crowdsourced dataset of street crime to analyze and predict that community-level and location-level features are significant sources of predictive signal for street crime severity. Comparative and/or case studies based on algorithmic approaches are less prevalent than single-model case studies, as stated by Kounadi et al. [12] in the systematic review of the methods used for spatial crime forecasting.

The reviewed studies listed above are all aimed at different types of tasks such as crime type prediction, hotspot detection, and severity forecasting, but they all show the potential of using machine learning techniques for crime related classification. Combined, these works call for a concentrated comparison of lazy learning algorithms, as previous studies do not selectively contrast this family of learning algorithms against a carefully designed and controlled experimental protocol for community-level violent crime classification.

3. METHODOLOGY

This section introduces the dataset and the pre-processing pipeline, along with the lazy learning algorithms and experimental settings for the construction and the test of the comparative models. Fig. 1 shows the entire pipeline, from dataset to final evaluation.

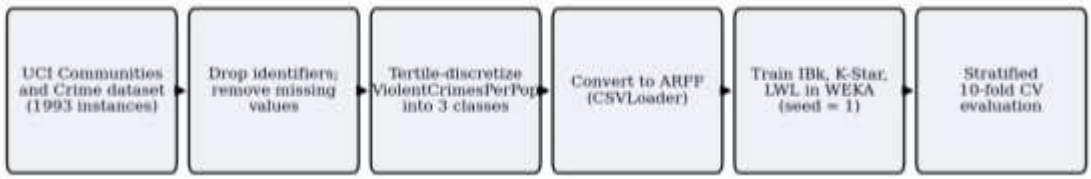

Figure 1: Experimental workflow from raw dataset to evaluated models

3.1. Dataset

This study used the Communities and Crime dataset, originally developed by Redmond and Baveja [13] and available from the UCI Machine Learning Repository [14]. It uses socio-economic data from the 1990 United States Census and Law Enforcement Management and Administrative Statistics (LEMAS) survey data from 1990, as well as data from the 1995 FBI Uniform Crime Report (UCR). The cleaned dataset contains 1,993 community instances after community records are removed with missing attribute values. Once the non-predictive identifier fields (state, community name, and cross-validation fold) are eliminated, 100 predictive numeric attributes are left.

These 100 attributes have been sorted into seven categories in Table I, according to the matching of the attribute names with the attributes described in the dataset documentation. These seven categories are not officially defined by the UCI, but were assigned as a short-hand way to summarize the composition of the predictor set by matching attribute name substrings, like Pct, Inc, or Lemas, to the thematic descriptions in the dataset documentation. Housing conditions are the largest category, followed by income and poverty, and immigration-related attributes. The missing-value cleaning step leaves one attribute in the original Law Enforcement Management and Administrative Statistics survey: the percentage of officers assigned to drug units. A high percentage of communities had missing data for many policing-related fields and these were omitted from the raw data. This lack of equally distributed attribute categories is mentioned here as a possible factor constraining the extent to which the models can be used to gain law-enforcement-specific signal.

The target variable ViolentCrimesPerPop is the number of violent crimes per 100,000 people, calculated from the number of murder, rapes, robberies and assaults. The present study formulates community-level violent crime prediction as a classification problem instead of a regression problem, so the continuous target is rewritten as a 3-part classification using equal-frequency (tertile) binning of the sorted target values. Consequently, the retrieved class

boundaries are 0.09 and 0.25 on the normalized 0.00–1.00 scale of this dataset. Several communities have the same target values at each of these boundaries, resulting in an almost balanced split: 679 Low, 661 Medium, and 653 High. This near balance is enough to prevent class imbalance issues that might create difficulties in comparison between classifiers.

Table i. Dataset attribute category summary (100 predictors)

Attribute category	**Count**	**% of total**
Housing characteristics	29	29.0%
Population & age/race demographics	14	14.0%
Income & poverty	17	17.0%
Education & employment	9	9.0%
Family & marital structure	13	13.0%
Immigration & language	17	17.0%
Law enforcement / policing	1	1.0%
Total	**100**	**100%**

3.2. Preprocessing

The discretization of ViolentCrimesPerPop into Low, Medium and High was performed in Python before the data was imported into WEKA, not through the supervised Discretize filter in WEKA, so that the boundaries of the discretization can be determined deterministically from the sorted target distribution. This resulted in feature and class columns being exported as a comma-separated file with the class attribute as the final column.

The processed dataset was subsequently converted to WEKA's native format (Attribute-Relation File Format) with the help of the CSVLoader utility provided in weka.core.converters, using default settings (infer nominal from 3 string values, numeric from everything else); it infers the class attribute as nominal, and everything else as numeric. The ARFF file created contains 100 numeric attributes and one nominal class attribute called CrimeLevel with the values Low, Medium and High. In addition to the normalization done on the original dataset, no additional normalization, attribute selection, nor feature scaling were performed to isolate the effect of the classification algorithm from the effect of feature engineering.

3.3. Lazy Learning Algorithms

Each lazy learning family was tested with the configuration of parameters as reported by WEKA at run time, which is reported below for reproducibility. The first is IBk, which is WEKA's version of the k-Nearest Neighbor algorithm that was first introduced by Cover and Hart [15] and then formalized as an instance-based learning framework by Aha, Kibler and Albert [16]. An IBk classifier finds the k nearest neighbors in the training set using Euclidean distance over the feature vector, and then uses

majority voting to determine the class of the query instance. To consider the sensitivity of accuracy with respect to neighborhood size, this study compares the accuracy of IBk with k = 1, 3, 5, and 7 (WEKA option -K) without any window size limit (WEKA option -W 0) and without any distance weighting (WEKA option -W 0).

K-Star [17] is the second algorithm proposed by Cleary and Trigg that uses an entropy-based similarity measure based on the probability of transforming one instance to another by a random sequence of elementary operations, instead of the Euclidean distance measure. The blending parameter was set to its default value (20, option -B 20) and missing values were replaced by the average column-value method (option -M a) which is the default setting of WEKA for this classification.

Locally Weighted Learning is the third algorithm studied, based on the algorithm framework authored by Atkeson, Moore, and Schaal [18]. LWL learns a base learner for each query point's neighborhood consisting of training instances. The default settings utilized throughout were: base classifier DecisionStump; linear weighting kernel; and all training neighbors at each query, in this case k = infinity. As for classification, LWL is designed for numeric prediction and for simple decision-stump style classification, and can serve as a useful contrast case with the two instance-voting methods, IBk and K-Star.

3.4. Experimental Setup and Statistical Testing

WEKA 3.6.14, an open-source machine learning workbench in the Java programming language, was used for all of the experiments, running on OpenJDK 21.0.10 on Ubuntu 24.04.4 LTS. To obtain a reliable estimate of model performance, stratified 10-fold cross-validation with random seed 1 (the default value for WEKA) was used, as suggested by Kohavi [19] for the reliable estimation of accuracy for classifiers with a small number of samples. In the multi-class setting used here, the number of Low, Medium, and High instances is preserved in each fold during stratification.

Six evaluation metrics were calculated for each classifier: Overall accuracy, Weighted precision, Weighted recall, Weighted F-measure, Kappa statistic, and Weighted area under the ROC curve. Confusion matrices were also taken to uncover error patterns by class, a type of error that may be lost in the summary metrics, especially for the Medium class, which is expected to have some overlap between the Low and High classes.

In order to enable a paired statistical comparison, the same partition with seed=1 was also materialized into ten explicit train/test file pairs using weka filters supervised instance Stratified Remove Folds. The three algorithms were each retrained and retested on all 10 identical fold pairs yielding 10 accuracy scores for each algorithm at its most successful configuration or at its default configuration. The ten folds are comprised of 199–200 test instances. Consequently, the unweighted average of the ten per-fold accuracies may be a few hundredths of a percentage point off from the single pooled accuracy from the full 10-fold run. Both are reported in Table II for clarity and all significance testing is based on the mean of the matched folds.

Cross-validation folds do not satisfy the assumption of independent samples as is required for ordinary paired t-test. To overcome this, the Nadeau and Bengio used the ratio of the test-set size to training-set size to increase the variance estimate when using the corrected resampled t-test in addition to the standard paired t-test and the Wilcoxon signed-rank test. Effect sizes were also calculated for paired samples using Cohen's d effect size and 95% confidence intervals of mean accuracy were calculated using the t distribution with 9 degrees of freedom. This comparison was with the intent of determining if the differences in accuracy reported in Section IV were statistically significant, and practically meaningful, as opposed to fold-to-fold variability, when comparing IBk with k = 7 to K-Star and to LWL.

4. RESULTS AND DISCUSSION

Table II shows results for the six model configurations. It displays mean accuracy and standard deviation over the 10 matched cross-validation folds (see Section III), as well as weighted precision, recall, F-measure, Kappa statistic and ROC area for the entire 10-fold run. IBk with k = 7 achieved the highest mean accuracy at 64.53% (SD = 2.87), closely followed by IBk with k = 5 at 64.22% (SD = 3.99). Both configurations achieved a higher score than IBk with k = 1, K-star, and LWL, all of which scored between 58.96% and 59.50%. These results are represented graphically in Fig. 2 (shown with error bars).

Table 2. Comparative performance of lazy learning algorithms (10-fold cv)

Classifier	**Accuracy (%) mean ± SD**	**95% CI**	**Precision**	**Recall**	**F-Meas.**	**Kappa**	**ROC**
IBk (k = 1)	59.21 ± 2.93	57.11–61.31	0.590	0.592	0.591	0.388	0.696
IBk (k = 3)	62.22 ± 2.29	60.58–63.86	0.618	0.622	0.618	0.433	0.777
IBk (k = 5)	64.22 ± 3.99	61.37–67.08	0.641	0.642	0.640	0.463	0.808
IBk (k = 7)	64.53 ± 2.87	62.47–66.58	0.643	0.645	0.643	0.468	0.816
K-Star (b = 20)	59.50 ± 3.39	57.08–61.93	0.600	0.595	0.597	0.392	0.780
LWL (D-Stump)	58.96 ± 1.81	57.66–60.25	0.400	0.590	0.474	0.383	0.797

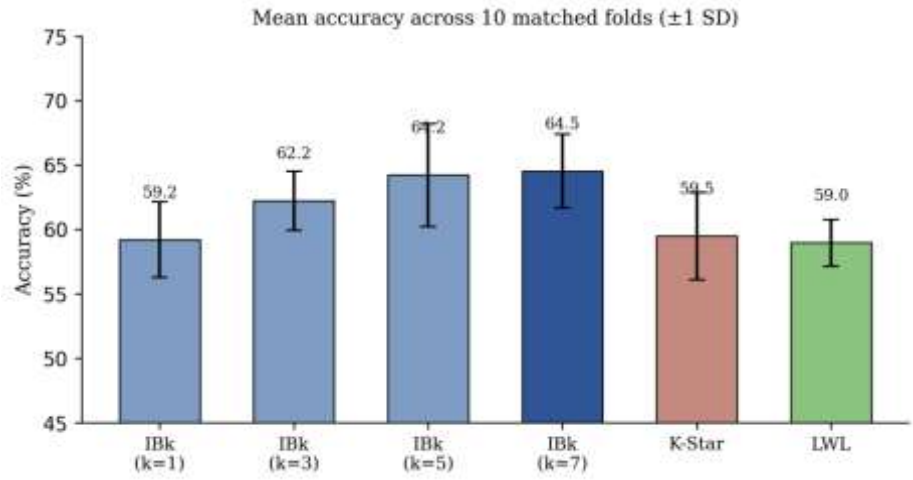

Figure 2. Mean accuracy across the ten matched cross-validation folds, with error bars showing one standard deviation.

Three complementary tests were used on the 10 per-fold accuracy scores to ensure that the accuracy advantage observed with IBk, k = 7, was not just a random fold-to-fold phenomenon. These include a standard paired t-test, a variance corrected resampled t-test that considers partial dependence between the cross-validation folds, and a Wilcoxon signed rank test as a non-parametric test. These tests were evaluated along with Cohen's d, which was used to measure effect size. All four statistics for the two comparisons of interest are given in Table III.

Table 3. Significance tests and effect sizes against ibk (k = 7), n = 10 matched folds

Comparison	**Paired t (p)**	**Corrected resampled t (p)**	**Wilcoxon p**	**Cohen's d**
IBk (k=7) vs. K-Star	t=5.38 (0.0004)	t=3.71 (0.0049)	0.0020	1.70
IBk (k=7) vs. LWL	t=5.59 (0.0003)	t=3.85 (0.0039)	0.0020	1.77

The corrected resampled t-test is the most conservative of these tests and yet both of its comparisons are still significant at the 0.01 level following the correction for fold dependence, confirming the rejection of the null hypothesis of no difference for all three tests. In both cases, Cohen's d is greater than 1.7, which is a large effect size and not a marginal one. When considered together, these results suggest that the improvement in accuracy of IBk with k = 7 over K-Star and LWL is statistically significant and practically meaningful, and not an artifact of the sampling variation of the folds.

A clear trend is observed across the IBk configurations. The mean accuracy steadily increased from k = 1 (59.21%) to k = 7 (64.53%) and the Kappa statistic steadily went up from k = 1 (0.388) to k = 7 (0.468). These findings suggest that the average over a moderate number of neighbors reduces sensitivity to noisy individual votes. This behavior may be attributed to the socio-economic factors of the predictors, as there is likely to be some variation in the reported crime rates of a similar community.

Mean accuracy and weighted F-measure of K-Star are 59.50% and 0.597, respectively, which is similar to IBk with k = 1. Theoretically, K-Star's distance measure based on entropy is more flexible than Euclidean, but this did not translate to the current dataset, perhaps because most of the attributes were already continuous and normalized before applying the distance measure.

While LWL achieved a similar weighted recall of 0.590, the overall accuracy was quite low at 58.96% and the weighted Precision was even lower at 0.400.

An examination of the per-class results presented in Table IV for the best model (IBk with k = 7) and the complete confusion matrices in Table V reveals this limitation. LWL was based on a DecisionStump base learner. LWL never classified a test instance as Medium, and classified all instances in the class Medium as Low or High. This is a natural consequence of the limited expressive power of a single-attribute decision stump, which cannot express the boundaries of an intermediate class that contains both the adjacent classes on several correlated attributes. By contrast, K-Star's confusion matrix does fill in the Medium class, though with a much greater amount of confusion with both adjacent classes than the IBk with k = 7.

The per-class metrics of the best-performing model, IBk with k = 7, are given in Table IV. The precision and recall values for the Low class were above 0.75 and 0.77 respectively, while for the High class they were above 0.75 and 0.76 respectively. For the Medium class, they were 0.769 and 0.608 respectively, and it was still the most difficult to separate. This is similar to the intuitive notion that communities with well-defined rates of violent crime – both low and high – can be better described by their socio-economic and policing characteristics than communities with rates of crime in the middle of the range.

Table 5. Per-class performance of the best model (ibk, k = 7)

Class	**TP Rate**	**FP Rate**	**Precision**	**Recall**	**F-Measure**	**ROC Area**
Low	0.850	0.145	0.751	0.850	0.798	0.935
Medium	0.608	0.150	0.668	0.608	0.637	0.830
High	0.775	0.087	0.812	0.775	0.793	0.938
Weighted Avg.	**0.645**	**0.128**	**0.644**	**0.645**	**0.643**	**0.902**

In general, the outcomes indicate that simple instance-based learning, based on neighbor votes (IBk) performs significantly better than entropy-based or kernel-weighted lazy learning for the community-level violent crime classification task at least for the neighborhood sizes and default configuration used in this study. This performance advantage is accompanied by a relatively strong ROC area (0.816), notwithstanding the comparatively low raw accuracy, since the middle class is an overlapping class in an inherently three-class classification problem.

Table 6. Confusion matrices for ibk (k = 7), k-star, and lwl (10-fold cv)

Model / Actual	**Pred. Low**	**Pred. Medium**	**Pred. High**
IBk (k = 7)			
Actual Low	515	146	18
Actual Medium	208	312	141
Actual High	32	162	459
K-Star (blend = 20)			
Actual Low	465	184	30
Actual Medium	193	313	155
Actual High	47	198	408
LWL (DecisionStump)			
Actual Low	636	0	43
Actual Medium	419	0	242
Actual High	114	0	539

However, the results should be considered in the context of some caveats. First, the dataset consolidates crime data at the community rather than street segment or crime activity level, which means that the predictions will refer to the risk of violent crime in a community, not a specific street segment, or even specific crime activities. Second, the equal-frequency binning was the conventional approach for transforming a continuous crime rate into near-balanced classes, but there is some class-boundary sensitivity that might lead to different comparative rankings if other binning approaches were used. Third, only one law-enforcement attribute remains after the missing value cleaning step – which is likely to reduce the amount of policing-specific signal that can be exploited by any model, not just the lazy learners tested here. Fourth, all data originate from a specific historical period, in this case 1990 census, 1990 LEMAS, and 1995 FBI records, so the reported accuracy is relative accuracy for this snapshot of data, and not a guaranteed accuracy for data from more recent or geographically different data.

6. CONCLUSION AND FUTURE WORK

This study compared three lazy learning algorithms, namely IBk, K-Star and Locally Weighted Learning, for predicting community-level violent crime categories as an operational measure of street crime. The publicly available Communities and Crime dataset was used for

experiments, along with the WEKA machine learning workbench. The best overall performance was obtained by using IBk (k = 7), which obtained a mean accuracy of 64.53%, a weighted F-measure of 0.643 and a Kappa statistic of 0.468, with 10-fold cross-validation repeated ten times on ten matched sets of folds. A corrected resampled t-test and a Wilcoxon signed rank test confirmed that IBk significantly outperformed both K-Star and LWL, with large effect size (Cohen's d > 1.7) in both cases.

Moreover, the comparison also uncovered a key qualitative disparity between algorithm families. The local models based on decision stumps were completely unsuccessful on the Medium crime category, while the neighbor-voting mechanism in IBk was able to deal with all three categories, although the Medium class was still the most challenging for all the models that were tested. This result has demonstrated that the choice of algorithm among lazy learners should not be based solely on predictive accuracy, but also on the way the method deals with the overlapping or intermediate classes as well. Future work may extend this comparison in a number of ways. The 100-attribute space could be reduced using feature selection, the IBk variants of distance-weighted could be assessed as in Yunus and Loo [11] and spatio-temporal crime data could be added to the data as in Kounadi et al. [12]. However, the lazy learner could be compared with ensemble methods, such as the stacked generalization method proposed by Kshatri et al. [9]. If the community-level patterns could be confirmed by incorporating additional crime data (at the street-segment-resolution) and more recent census and crime data at the city level, the results would provide additional evidence of whether these patterns extend to finer-grained and more recent street crime prediction tasks.

6. REFERENCES

[1] I. H. Witten, E. Frank, and M. A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, 3rd ed. Burlington, MA, USA: Morgan Kaufmann, 2011.

[2] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, 3rd ed. Burlington, MA, USA: Morgan Kaufmann, 2011.

[3] R. Iqbal, M. A. A. Murad, A. Mustapha, P. H. Shariat Panahy, and N. Khanahmadliravi, "An experimental study of classification algorithms for crime prediction," Indian Journal of Science and Technology, vol. 6, no. 3, pp. 4219–4225, Mar. 2013, doi: 10.17485/ijst/2013/v6i3.6.

[4] S. Kim, P. Joshi, P. S. Kalsi, and P. Taheri, "Crime analysis through machine learning," in Proc. 9th IEEE Annual Information Technology, Electronics and Mobile Communication Conf. (IEMCON), Vancouver, Canada, 2018, pp. 415–420, doi: 10.1109/IEMCON.2018.8614828.

[5] A. H. Wibowo and T. I. Oesman, "The comparative analysis on the accuracy of k-NN, naive Bayes, and decision tree algorithms in predicting crimes and criminal actions in Sleman regency," Journal of Physics: Conference Series, vol. 1450, art. 012076, 2020, doi: 10.1088/1742-6596/1450/1/012076.

[6] S. A. Chun, V. A. Paturu, S. Yuan, R. Pathak, V. Atluri, and N. R. Adam, "Crime forecasting: A machine learning and computer vision approach to crime prediction and prevention," Visual Computing for Industry, Biomedicine, and Art, vol. 4, art. 9, 2021, doi: 10.1186/s42492-021-00075-z.

[7] M. Khan, A. Malik, and J. Ahmed,

"Predicting and preventing crime: A crime prediction model using San Francisco crime data by classification techniques," Complexity, vol. 2022, art. 4830411, 2022, doi: 10.1155/2022/4830411.

[8] V. Mandalapu, L. Elluri, P. Vyas, and N. Roy, "Crime prediction using machine learning and deep learning: A systematic review and future directions," IEEE Access, vol. 11, pp. 60153–60170, 2023, doi: 10.1109/ACCESS.2023.3286344.

[9] S. S. Kshatri, D. Singh, B. Narain, S. Bhatia, M. T. Quasim, and G. R. Sinha, "An empirical analysis of machine learning algorithms for crime prediction using stacked generalization: An ensemble approach," IEEE Access, vol. 9, pp. 67488–67500, 2021, doi: 10.1109/ACCESS.2021.3075140.

[10] U. M. Butt, S. Letchmunan, F. H. Hassan, M. Ali, A. Baqir, and H. H. R. Sherazi, "Spatio-temporal crime hotspot detection and prediction: A systematic literature review," IEEE Access, vol. 8, pp. 166553–166574, 2020, doi: 10.1109/ACCESS.2020.3022808.

[11] A. Yunus and J. Loo, "London street crime analysis and prediction using crowdsourced dataset," Journal of Computational Mathematics and Data Science, vol. 10, art. 100089, 2024, doi: 10.1016/j.jcmds.2023.100089.

[12] O. Kounadi, A. Ristea, A. Araujo, and M. Leitner, "A systematic review on spatial crime forecasting," Crime Science, vol. 9, art. 6, 2020, doi: 10.1186/s40163-020-00116-7.

[13] M. Redmond and A. Baveja, "A data-driven software tool for enabling cooperative information sharing among police departments," European Journal of Operational Research, vol. 141, no. 3, pp. 660–678, 2002, doi: 10.1016/S0377-2217(01)00264-8.

[14] M. Redmond, "Communities and Crime," UCI Machine Learning Repository, 2002. [Online]. Available: https://doi.org/10.24432/C53W3X.

[15] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," IEEE Transactions on Information Theory, vol. 13, no. 1, pp. 21–27, Jan. 1967, doi: 10.1109/TIT.1967.1053964.

[16] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," Machine Learning, vol. 6, no. 1, pp. 37–66, Jan. 1991, doi: 10.1007/BF00153759.

[17] J. G. Cleary and L. E. Trigg, "K*: An instance-based learner using an entropic distance measure," in Proc. 12th Int. Conf. Machine Learning, Tahoe City, CA, USA, 1995, pp. 108–114.

[18] C. G. Atkeson, A. W. Moore, and S. Schaal, "Locally weighted learning," Artificial Intelligence Review, vol. 11, no. 1–5, pp. 11–73, Feb. 1997, doi: 10.1023/A:1006559212014.

[19] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in Proc. 14th Int. Joint Conf. Artificial Intelligence (IJCAI), Montreal, Canada, 1995, pp. 1137–1143.

[20] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," ACM SIGKDD Explorations Newsletter, vol. 11, no. 1, pp. 10–18, Nov. 2009, doi: 10.1145/1656274.1656278.