Journal of Theoretical and Applied Information Technology
31st December 2021. Vol.99. No 24

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

EXPERIMENTAL ANALYSIS OF HYPERPARAMETERS TUNING ON CLASSIFICATION OF CRIME NEWS IN INDONESIA

[1]AJULIO PADLY SEMBIRING, [2]SHARFINA FAZA, [3]MAHARDIKA ABDI PRAWIRA TANJUNG

[1,2] Department of Computer and Informatics Engineering, Politeknik Negeri Medan, Indonesia

[3]Department of Informatics Engineering, Sekolah Tinggi Teknologi Sinar Husni, Medan, Indonesia

E-mail: mahardhikaprawira@gmail.com

ABSTRACT

Patterns and trends of criminal acts can be quantified with classification and statistics. This classification uses criminal news headlines from online news media. Every day, online news media post illegal news. It simplifies the classifying procedure. Criminal patterns and trends can be easily identified if law enforcement organizations categorize criminal events based on online news sources. According to previous research, using the correct methodologies and parameters to define illegal activities is difficult in Indonesia. Researchers will utilize a hyperparameter tuning scheme on KNN, Random Forest, and SVM methods to determine the optimal techniques and parameters for the criminal act dataset. The support vector machine is the best classifier for this investigation, both in model and hyperparameter tweaking. In the fourth test, the accuracy value declined by -0.52 percent, and in the fifth test, the accuracy value decreased by -0.52 percent. The accuracy value is -0.15%, and the sixth test reduced it by -0.26%. This decline is because the support vector machine parameter cannot classify string vector data. The random forest classifier gains the most accuracy with hyperparameter adjustment (1.74%).

Keywords: *Analysis, Classification, Crime, Experimental, Hyperparameter*

1. INTRODUCTION

Crime acts are increasing rapidly from day to day and have significant social effects, so crime is a major issue that continues to grow [1]. Crime is a form of action that violates civil and criminal law, one of society's dominant and disturbing issues [2]. Law enforcement agencies obtain crime data from various sources: the existing online information media [3]. Crime is difficult to predict, and from the data collection, it was found that many factors influence crime, such as unemployment, poverty, and drugs [4].

In handling and preventing criminal acts, law enforcement agencies should find patterns of criminal acts [5]. However, this becomes difficult to do when relying on conventional methods, and the law enforcement agencies must take advantage of advances in information technology. This handling and preventing criminal acts need to use machine learning to find patterns of existing criminal acts [6].

The growing need to mitigate crimes gave rise to this research work by applying machine learning techniques from the data from Indonesian crime news to break down the different crimes and build a model to classify these crimes. Looking at the other models as far as execution to check how well the crimes were classified and help the government and law enforcement agencies return [7]. To get an insight into the most common type of crimes, they encounter daily and enable them to take careful steps to overcome these criminal activities.

Machine learning algorithms were used to develop a classification model trained to analyze the crime rate [8]. Linear Regression, KNN, Naïve Bayes, Random Forest, Support Vector Machine, etc., are some of the classification machine learning algorithms that various researchers have deployed over the years to classify crimes using a dataset. One of the challenges machine learning faces in crime analysis is accuracy [8]. The result showed that their model had a low prediction rate, which did not perform as expected [8]–[11].

The low prediction rate that previous works resulted in may cause high noise and outliers on the classifiers they used. The increased noise and outliers will reduce the processing output due to the randomness property of each classifier used. One better approach to improve the outcome of any classifier is to tune the hyperparameters of that classifier [12], [13]. The parameters set by the data analysts before the training process are called hyperparameters and are independent of the training process. For example, in a random forest, a hyperparameter would be how many trees have to be included in the forest or how many nodes each tree can have. Optimizing these hyperparameters for the classifier is the key to the perfect prediction of unlabeled data [14]. These can only be achieved through trial and error methods. Different values of hyperparameters are used, then compare their result and finally find the best combination of them. The tuning process of hyperparameters is mainly dependent on experimental results.

This work applies default model tuning and hyperparameter tuning approach to tuning the K-nearest neighbor (KNN), Support Vector Machine (SVM), and Random Forest classifier to identify the best parameter on each classifier and analyze it. The implementation of Hyperparameter tuning is simple [15]–[17]. A set of hyperparameters and their values are fed to it first. Then run a complete search overall for all possible combinations of given values, then train the model for each set of values. Then hyperparameter tuning will compare the score of each model it introduces and keep the best one. A common extension of hyperparameter tuning is to use cross-validation, training the model on several folds with different hyperparameter combinations to find more accurate results.

2. MATERIALS AND METHODS

2.1 Dataset

The dataset used is a collection of criminal news from okezone.com, kompas.com, tribun.com, and detik.com. The dataset results from data crawling with the query "criminal news" labeled based on the type of crime. Then the data is collected into one excel file named "headline_crime_dataset.csv" which consists of 9,000 rows of real headline, date, news type, and tag data. The labeling given to the criminal news headlines is criminal news headlines in the category of theft with 3,000 data lines, narcotics with 3,000 data lines, and murder with 3,000 data lines. The columns used in this study are only the headline and tag column because the headline contains the main data, and the tag contains the labeling of the crime category. Here is how headline_crime_dataset.csv looks like:

	Headline	Date	NewsType	Tag
2	Remaja Tak Jera Dipenjara Malah Bikin Teller Bank Hilang Nyawa	12/31/2020 22:02	detikNews	pencurian
3	Amati Lampu yang Nyala Siang Bolong, Rampok Ini Bobol 36 Rumah	12/31/2020 20:41	detikNews	pencurian
4	Sepanjang 2020, Kasus Penipuan-Pencurian Tertinggi Terjadi di Medan	12/31/2020 18:37	detikNews	pencurian
5	Aksi Penembakan KKB di Mimika Meningkat Tajam di 2020, Ini Sebabnya	12/31/2020 15:58	detikNews	pencurian
6	Tusuk Korban Berkali-kali, Ini Motif ABG Bunuh Karyawati Bank di Denpasar	12/31/2020 14:23	detikNews	pencurian
7	Ini 10 Kasus Pidana Paling Menonjol di Sukabumi Selama 2020	12/31/2020 12:43	detikNews	pencurian
8	Pembunuh Karyawati Bank di Denpasar Residivis Pencurian Kotak Sesari	12/31/2020 10:34	detikNews	pencurian
9	Misteri Tewasnya Teteh Penjual Jamu-3 Kasus yang Belum Terungkap di Garut	12/31/2020 10:16	detikNews	pencurian
10	Tren Kejahatan Menonjol di Jakbar di 2020: Pencabulan-Penyelundupan Narkoba	12/31/2020 8:48	detikNews	pencurian
11	Terbongkar Pencurian di Soetta Libatkan Warga Afghanistan Pencari Suaka	12/31/2020 6:50	detikNews	pencurian
12	367 Pelaku Kejahatan di NTB Ditangkap Selama 2 Pekan	12/31/2020 0:28	detikNews	pencurian
13	Residivis Curanmor Ditangkap Polisi saat Hendak Curi Kotak Amal Masjid	12/31/2020 0:07	detikNews	pencurian
14	Warga Kota Malang Ini Rampas 94 HP Bermodus Razia Masker	12/30/2020 21:21	detikNews	pencurian
15	Kasus Curanmor di Surabaya Naik Jika Dibanding 2019	12/30/2020 19:08	detikNews	pencurian
16	Kasus Curat dan Penipuan Tertinggi di Tuban Selama 2020	12/30/2020 18:24	detikNews	pencurian
17	Beli Laptop Hasil Curian di Bandara Soetta, WN Afganistan Ikut Ditangkap	12/30/2020 16:21	detikNews	pencurian
18	Polisi Tangkap Pelaku Pencurian Laptop Penumpang di Bandara Soetta	12/30/2020 15:08	detikNews	pencurian
19	Curi Ikan di Kepri, 5 Kapal Nelayan Vietnam-Malaysia Ditenggelamkan	12/30/2020 14:21	detikNews	pencurian

Figure 1: Display of "headline_crime_dataset"

2.2 Preprocessing Data

In the preprocessing stage of the dataset used, it is carried out with the following steps:

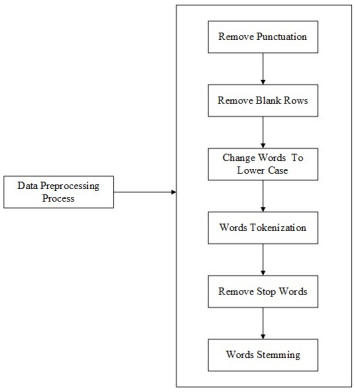

Figure 2: Data Preprocessing Stages

At the beginning of the data preprocessing stage, punctuation marks are removed from the words in the news headlines, then deleted if there are empty lines in the dataset, followed by changing the letters to all lowercase letters. The words in the news headline sentences are broken down into units. Vector is like breaking the word "I am the mind," then it will become "I" into vector number 1 "is" becomes vector number two "Budi" becomes vector number 3. The next step is to eliminate stop words, which is to eliminate words that are very commonly used, so it brings only a little useful information. The last step is to do word stemming, which removes affixes, prepositions, and conjunctions. In contrast to the word-stemming of English words which only removes suffixed words, Indonesian word-stemming eliminates suffixes and prefixes.

2.3 Model Tuning

After performing the data preprocessing stage, the data classification stage will be carried out with three different classifier methods. This step is intended to find classification accuracy; here are the classification stages:

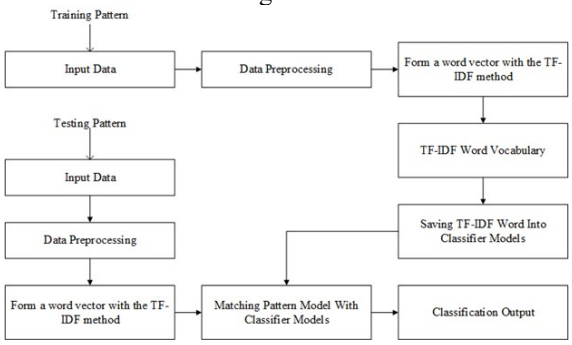

Figure 3: Model Classification Stages

2.3.1 K-Nearest Neighbor (KNN) Classifier

The K-nearest neighbor (KNN) algorithm is a type of instance-based learning. Additionally, this algorithm is a technique for lazy learning. KNN is performed by identifying k objects in the training data that are most similar (identical) to the new or testing data [18]. For instance, it is useful to identify a new data problem utilizing a solution from previous data. We employ proximity to the old data case to identify a solution from the new data; the solution from the old case closest to the new case is used as a solution.

As illustrated in Figure 4, four new data sets and four older data sets, namely P, Q, R, and S. When new data becomes available, the closest old data case is used as the answer.

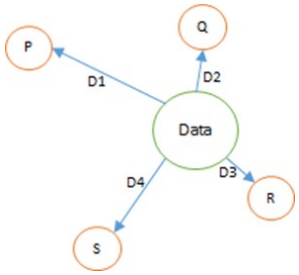

Figure 4: K-nearest neighbor case illustration

D1 represents the distance between new data and P data, D2 represents the distance between new data and Q data, D3 represents the distance between new data and R data, and D4 represents the distance between new data and S data. With the addition of a new case. Thus, the solution for the Q data case will be applied to the new data.

There are numerous methods for determining the proximity of new and old data (training data), including the Euclidean and Manhattan distances (city block distances), but the most frequently used is the Euclidean distance [19], which is as follows:

$$\sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \cdots + (a_n - b_n)^2} \quad (1)$$

Where a = a1, a2, …, an, and b = b1, b2, …, bn represents n attribute values.

2.3.2 Support Vector Machine (SVM) Classifier

The support vector machine converts the original training data to higher dimensions via nonlinear mapping [20]. This new dimension searches for a linear hyperplane, the optimal separator, or by definition is a "decision boundary" that separates data from one class from another. With sufficient high-dimensional nonlinear mapping, the data of the two classes are separated by a hyperplane. Using support vectors (class boundaries) and margins (defined by support vectors), SVM determines this hyperplane [21].

The support vector machine looks for the maximum margin distance from the hyperplane to separate two different classes. The support vector machine can be illustrated as follows:

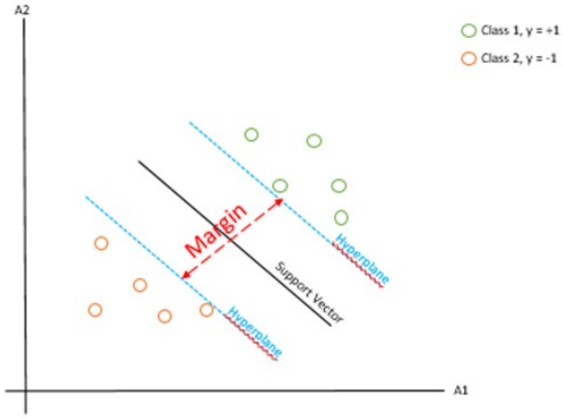

Figure 5: Support vector machine dan the hyperplane (adapted from [21]).

The weights can be modified so that the hyperplane encompasses the existing training data's margins. The formulation can be written as follows [13].

H_1: w_0+ w_1 x_1+ w_2 x_2≥+1 for y_i= +1 (2), and
H_2: w_0+ w_1 x_1+ w_2 x_2≤-1 for y_i= -1 (3)

Therefore, data equal to or above H_1 belongs to class +1, and any data equal to or below H_2 belongs to class -1.

2.3.3 Random Forest Classifier

Random Forest is a classification algorithm that consists of more than one decision tree. Each decision tree is constructed using the values of a random vector sampled independently and uniformly across all trees [22]. Random Forest belongs to the Supervised Learning group developed by Leo Bremen. This method is one of the most accurate classification methods used in making predictions, can handle many input variables without overfitting, and helps eliminate correlations between decision trees such as usual ensemble methods [23]. The following is a methodology for how Random Forest works, as shown in Figure 6.

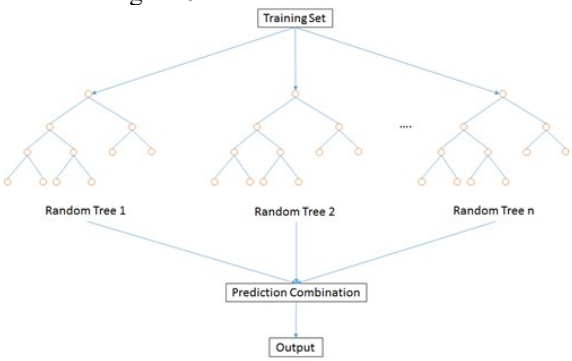

Figure 6: How Random Forest Work

In making a decision tree, a random forest uses a random vector. The pseudocode stages in making Random Forest [24]; the first step is to select feature "R" from the full feature "m" where R<<m, continued with among the "R" features, then count the vertices using the best split point. The next step is to splits a node into child nodes using the best split, then repeat steps a to c until "1" the number of nodes has been reached. The last step is to Build a forest by repeating steps a to d for "n" times to make the "n" number of trees.

The advantage of utilizing the Random Forest algorithm is a classification approach is that the random forest never encounters the overfitting problem. The Random Forest algorithm can do both regression and classification and identify the most significant features to employ from the training dataset [25].

2.4 Hyperparameter Tuning

Hyperparameters are parameters associated with the training method not discovered during the training process [26]. Hyperparameter tweaking is accomplished through the execution of several trials inside a single training job. Each trial is a complete execution of one's training application using the values for one's specified hyperparameters [26]. When the work is complete, the user can summarize all the trials and the most effective configuration of values based on the specified criteria. Hyperparameter tuning improves a single specified target variable, also known as the hyperparameter metric. A common metric is the model's correctness as determined by an evaluation pass. The metric must be a numeric value, and the model can be tuned to maximize or minimize the metric [26].

Research conducted by [27] uses grid search to find the best parameters in hyperparameter tuning in a classifier. By using grid search, the engine will test one by one the available parameters in a classifier systematically according to the parameters that have been set [27]. For this reason, in this study, an experimental test will be carried out using a grid search hyperparameter tuning. The stages of hyperparameter tuning carried out in this study are as follows:

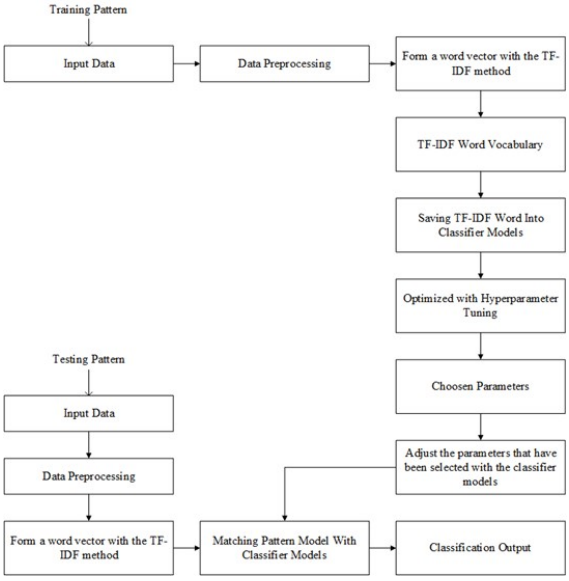

Figure 7: Hyperparameter Tuning Stages

The hyperparameter stages in Figure 7 are not much different from the tuning model stages previously described in section 2.3. There are several additional stages, including performing parameter optimization with hyperparameter tuning. This stage is carried out to find the best parameters to get the highest accuracy results. After the parameters are obtained, parameter value

adjustments are made on the models to be tested for their best accuracy. This stage is slightly different from the hyperparameter tuning stages carried out by other researchers [28], [29]. In general, other researchers apply the results of their tuning hyperparameters to the test data directly.

3. RESULT

The results discussed include model tuning, hyperparameter tuning, tuning computation time, and chosen parameters using the k-nearest neighbor classifier, support vector machine, and random forest. The test was performed ten times on the tuning model and hyperparameter tuning, with the steps described in sections 2.3 and 2.3.4. The results obtained in this study use the scikit-learn.org library and the Python programming language.

In the tuning model, the test uses the default parameter values of each classifier. The parameters selected for each classifier are summarized in table 1. By using hyperparameter tuning, these parameters are tested to get the best value for each parameter. In this way, the best values for the parameters in each classifier are obtained.

Table 1: Choosen Parameters

Classifier	Default Parameter	After Tuning	Tuning Computation Time
	Value		
KNN	n_neighbors=5, weights= uniform, algorithm = auto	n_neighbors=19 weights= distance, algorithm = auto	5.7 minutes
SVM	c=1, kernel = rbf, gamma = scale	C=1, Kernel = rbf, Gamma = scale	7.1 minutes
Random Forest	bootstrap=true, max_depth= none, max_features= auto, min_samples_leaf=1, min_samples_split=2, n_estimators=100	bootstrap= true, max_depth= None, max_features= sqrt, min_samples_leaf=2, min_samples_split=5, n_estimators=100	

After finding the best value for each classifier parameter, the classification of criminal acts was tested ten times due to the random distribution between training data and test data of 0.3 ratio. After ten tests, the final result is the average accuracy of each classifier using the tuning model and hyperparameter tuning. The test results are summarized in Table 2. Each classifier will record the accuracy of the tuning model, tuning hyperparameters, and the difference in accuracy between the two from each experiment carried out.

Table 2: Testing Result

Testing	KNN			SVM			Random Forest		
	M.T	H.T	A.D	M.T	H.T	A.D	M.T	H.T	A.D
Test 1	82.22	83.44	+1.22	88.30	89.30	+1.00	85.96	86.56	+0.60
Test 2	82.44	83.66	+1.22	88.41	88.89	+0.48	85.19	86.93	+1.74
Test 3	82.59	83.0	+0.41	88.52	88.74	+0.22	84.15	85.30	+1.15
Test 4	82.52	83.15	+0.63	88.78	88.26	-0.52	86.04	86.78	+0.74
Test 5	81.85	82.63	+0.78	88.96	88.81	-0.15	86.70	87.00	+0.30
Test 6	80.71	82.37	+1.66	88.78	88.52	-0.26	86.33	86.63	+0.30
Test 7	81.44	82.37	+0.93	88.19	88.70	+0.51	84.44	84.85	+0.41
Test 8	82.07	83.70	+1.63	87.93	88.11	+0.18	85.41	85.48	+0.07
Test 9	82.48	83.11	+0.63	88.30	88.70	+0.40	86.26	86.81	+0.56
Test 10	82.22	83.55	+1.33	87.33	87.81	+0.48	84.59	84.81	+0.22
Highest Accuracy	82.59	83.70	-	88.96	89.30	-	86.70	87.00	-
Lowest Accuracy	80.71	82.37	-	87.33	87.81	-	84.15	84.81	-
Average Accuracy	82.05	83.10	-	88.35	88.58	-	85.51	86.12	-
Highest A.D	-	-	+1.66	-	-	+1.00	-	-	+1.74

Note: M.T = Model Tuning, H.T= Hyperparameter Tuning, A.D= Accuracy Difference

4. DISCUSSION

After the testing stage, the results are as in table 2. The k-nearest neighbor gets the highest

accuracy value at the tuning model stage of 82.59%, the lowest accuracy is 80.71%, and the average accuracy obtained is 82.05%. While the support vector machine classifier gets the highest accuracy at the model tuning stage of 88.96%, the lowest accuracy is 87.33%, and the average accuracy obtained is 88.35%. The random forest classifier gets the highest accuracy at the model tuning stage of 86.70%, the lowest accuracy is 84.15%, and the average accuracy is 85.51%.

In the hyperparameter tuning stage, the k-nearest neighbor classifier gets the highest increase in accuracy with a value of 83.70%, the lowest accuracy with 82.37%, and an average increase in accuracy with a value of 83.10%. The support vector machine classifier gets the highest accuracy increase with 89.30%, the lowest accuracy with 87.81 %, and the average accuracy with 88.58%. The random forest classifier got the highest growth accuracy with 87.00%, the most insufficient accuracy with 84.81%, and a moderate increase in accuracy with 86.12%.

Interesting information obtained from the test results obtained on the support vector machine classifier is that the classification accuracy value does not always increase after hyperparameter tuning is done. Unlike the previous work that achieved improvisation of SVM classifier accuracy after hyperparameter tuning [30]–[32]. This information can be seen in the 4th, 5th, and 6th experiments. In the 4th experiment, performance decline inaccuracy of -0.52% was obtained, the 5th experiment performed a decline in accuracy of -0.15%, and the 6th experiment performed a decrease in accuracy of -0.26% was obtained.

This decline is due to the limitations of the support vector machine has, which has limitations in classifying string vector data types. These limitations achieved in this work may not be in line with what other researchers have succeeded in improving SVM accuracy [30]–[32]. The gamma parameter on the support vector machine can only accurately classify string vector data with 'scale' and 1. In contrast, the other values, namely the values of 0.1, 0.01, and 0.001 that were tested, produced a very low accuracy value, which was below 35%.

Meanwhile, the other two classifiers, k-nearest neighbor and random forest, consistently increased the classification accuracy value after hyperparameter tuning. The random forest classifier obtained the largest increase in accuracy among the other three classifiers. Which is +1.74 addition to the classification accuracy value, followed by k-nearest neighbor of +1.66 adding to the classification accuracy value, and the smallest is the support vector machine of +1.00 adding to the classification accuracy value.

5. CONCLUSION

In this study, the support vector machine is the highest accuracy classifier for classifying criminal acts, both in model tuning and hyperparameter tuning. Hyperparameter tuning is very influential in increasing the classification accuracy value of the random forest classifier. Still, hyperparameter tuning does not always increase the accuracy value for the support vector machine classifier. This accuracy decline is evidenced in the fourth test, and the accuracy value decreased by -0.52%, the fifth test decreased the accuracy value is -0.15%, and the sixth test has reduced by -0.26%. This decline is due to the parameter support vector machine's limitations, which have limitations in classifying string vector data types. Meanwhile, hyperparameter tuning provides the highest accuracy increase in the random forest classifier of +1.74%.

REFRENCES:

[1] M. Dulkiah, "Pengaruh Kemiskinan Terhadap Tingkat Tindak Kriminalitas Di Kota Bandung," *JISPO J. Ilmu Sos. dan Ilmu Polit.*, vol. 8, no. 1, pp. 36–57, 2018.

[2] R. S. Putra and Y. Kadarisman, "Kriminalitas Di Kalangan Remaja (Studi Terhadap Remaja Pelaku Pencabulan Di Lembaga Pemasyarakatan Anak Kelas Ii B Pekanbaru)." Riau University, 2016.

[3] F. N. Damayanti, I. N. Piarsa, and I. M. Sukarsa, "Sistem Informasi Geografis Pemetaan Persebaran Kriminalitas di Kota Denpasar," *Univ. Udayana, Denpasar*, 2016.

[4] M. Nisar, S. Ullah, M. Ali, and S. Alam, "Juvenile delinquency: The Influence of family, peer and economic factors on juvenile delinquents," *Appl. Sci. Reports*, vol. 9, no. 1, pp. 37–48, 2015.

[5] J. Mena, *Machine learning forensics for law enforcement, security, and intelligence*. CRC Press, 2016.

[6] M. L. Rich, "Machine learning, automated suspicion algorithms, and the fourth amendment," *Univ. PA. Law Rev.*, pp. 871–929, 2016.

[7] H. Chen *et al.*, "COPLINK Connect: information and knowledge management for law enforcement," *Decis. Support Syst.*, vol. 34, no. 3, pp. 271–285, 2003.

[8] S. Kim, P. Joshi, P. S. Kalsi, and P. Taheri, "Crime Analysis Through Machine Learning," *2018 IEEE 9th Annu. Inf. Technol. Electron. Mob. Commun. Conf. IEMCON 2018*, no. January, pp. 415–420, 2019, doi: 10.1109/IEMCON.2018.8614828.

[9] H. Gemasih, R. Rayuwati, A. SN, and M. Mursalin, "Classification of Criminal Crimes From Data Twitter Using Class Association Rules Mining," 2019, doi: 10.4108/eai.20-1-2018.2281925.

[10] Y. Lamari, B. Freskura, A. Abdessamad, S. Eichberg, and S. de Bonviller, "Predicting spatial crime occurrences through an efficient ensemble-learning model," *ISPRS Int. J. Geo-Information*, vol. 9, no. 11, 2020, doi: 10.3390/ijgi9110645.

[11] N. O. Edoka, "Crime Incidents Classification Using Supervised Machine Learning Techniques: Chicago," 2020, [Online]. Available: http://norma.ncirl.ie/4315/1/nelsonomonighoedoka.pdf.

[12] M. Yar, "The Novelty of 'Cybercrime' An Assessment in Light of Routine Activity Theory," *Eur. J. Criminol.*, vol. 2, no. 4, pp. 407–427, 2005.

[13] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," *arXiv Prepr. arXiv1801.06146*, 2018.

[14] X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer, "S4l: Self-supervised semi-supervised learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1476–1485.

[15] S. Maghfiroh, S. Basuki, and Y. Azhar, "Klasifikasi Tweets Tindak Kejahatan Berbahasa Indonesia Menggunakan Naive Bayes," *J. Repos.*, vol. 2, no. 7, pp. 933–944, 2020.

[16] S. S. Keerthi, "Efficient tuning of SVM hyperparameters using radius/margin bound and iterative algorithms," *IEEE Trans. Neural Networks*, vol. 13, no. 5, pp. 1225–1229, 2002.

[17] S. Falkner, A. Klein, and F. Hutter, "BOHB: Robust and efficient hyperparameter optimization at scale," in *International Conference on Machine Learning*, 2018, pp. 1437–1446.

[18] I. Gazalba and N. G. I. Reza, "Comparative analysis of k-nearest neighbor and modified k-nearest neighbor algorithm for data classification," in *2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, 2017, pp. 294–298.

[19] M. Bramer, *Principles of data mining*, vol. 180. Springer, 2007.

[20] H. Su, Z. Chen, and Z. Wen, "Performance improvement method of support vector machine-based model monitoring dam safety," *Struct. Control Heal. Monit.*, vol. 23, no. 2, pp. 252–266, 2016.

[21] J. Han, M. Kamber, and D. Mining, "Concepts and techniques," *Morgan Kaufmann*, vol. 340, pp. 93205–94104, 2006.

[22] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[23] T. N. Nuklianggraita, A. Adiwijaya, and A. Aditsania, "On the Feature Selection of Microarray Data for Cancer Detection based on Random Forest Classifier," *J. INFOTEL*, vol. 12, no. 3, pp. 89–96, 2020.

[24] W. Lin, Z. Wu, L. Lin, A. Wen, and J. Li, "An ensemble random forest algorithm for insurance big data analysis," *Ieee access*, vol. 5, pp. 16568–16575, 2017.

[25] S. Polamuri, "How The Random Forest Algorithm Works in Machine Learning," *dataaspirant.com*, 2018. https://dataaspirant.com/random-forest-algorithm-machine-learing/ (accessed Sep. 13, 2021).

[26] C. Google, "Overview of hyperparameter tuning," *cloud.google.com*, 2016. https://cloud.google.com/ai-platform/training/docs/hyperparameter-tuning-overview?cv=1 (accessed Sep. 13, 2021).

[27] B. H. Shekar and G. Dagnew, "Grid search-based hyperparameter tuning and classification of microarray cancer data," in *2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP)*, 2019, pp. 1–8.

[28] P. Liashchynskyi and P. Liashchynskyi, "Grid search, random search, genetic algorithm: a big comparison for NAS," *arXiv Prepr. arXiv1912.06059*, 2019.

[29] L. Zahedi, F. G. Mohammadi, S. Rezapour, M. W. Ohland, and M. H. Amini, "Search algorithms for automated hyper-parameter tuning," *arXiv Prepr. arXiv2104.14677*, 2021.

[30] B. M. Hsu, "Comparison of supervised classification models on textual data," *Mathematics*, vol. 8, no. 5, 2020, doi: 10.3390/MATH8050851.

[31] P. Probst, B. Bischl, and A.-L. Boulesteix, "Tunability: Importance of hyperparameters of machine learning algorithms," *arXiv Prepr. arXiv1802.09596*, 2018.

[32] Z. A. Sunkad, "Feature selection and hyperparameter optimization of SVM for human activity recognition," in *2016 3rd International Conference on Soft Computing & Machine Intelligence (ISCMI)*, 2016, pp. 104–109.