Neutrosophic Systems with Applications, Vol. 17, 2024
https://doi.org/10.61356/j.nswa.2024.17246

Software Reliability Model Estimation for an Indeterministic Crime Cluster through Reinforcement Learning

Dileep Kumar Kadali [1], R.N.V. Jagan Mohan [2,*], and M. Chandra Naik [1]

[1] Department of Computer Science and Engineering, GIET University, Gunupur, Odisha, 765022, India. Emails: dileep.kadali@giet.edu; srichandra2007@gmail.com.
[2] Sagi Rama Krishnam Raju Engineering College, Bhimavaram, 534204, India; mohanrnvj@gmail.com; mohan.rnvj@srkr.edu.in.

* Correspondence: mohan.rnvj@srkr.edu.in.

Abstract: The software reliability model estimates the probability of data failure in a specific environment, significantly impacting reliability and trustworthiness. The paper study focuses on cluster crime data, i.e., indeterministic in Neutrosophic Logic, using a software reliability model. The study utilizes reinforcement learning, Neutrosophic logic, and non-homogeneous Poisson process crime data to estimate indeterministic cluster data in crime. The "Non-homogeneous Poisson Process with Neutrosophic Logic" technique performs well in evaluating and deterring crime based on crime data analysis. The crime cluster involving offenders correctly classified as failure to accomplish does better than uncertain cluster reliability estimation with least squares and logistic regression analysis. The method enables crime prediction and prevention by using concave growth models to create an uncertain crime cluster, penalizing the correct person.

Keywords: Non-homogenous Poison Process; Neutrosophic Logic; Reinforcement Learning; Uncertain Crime Reliability Estimation.

1. Introduction

Crime clusters" are the tendency for crimes to congregate along the time, place, and other dimensions used to quantify them by Aparna [1]. Strategically, the ability to anticipate any crime based on timing, location, and other characteristics can help law enforcement by providing crucial information. Individuals with good self-discipline are more likely to commit crimes, while those with poor self-discipline are more likely to engage in illegal activities. A person has committed a crime when they blatantly violate the law through action, omission, or carelessness for which they may face punishment. A crime is an illegal act that violates a law or social standard, is punishable by law, and is approved by the government. Reliability refers to the consistency of measurement, ensuring results can be reproduced under the same circumstances [2]. While cluster integrity looks at the internal cohesion and separation of the clusters, cluster veracity assesses the external consistency and crime application of the clusters. The clustering analysis results can be accurate and beneficial when both variables are considered by J.A. Adeyiga [3]. Conducting a thorough investigation is crucial to determining if you are a party to the specific crime committed, as determining fault is challenging. Insufficient, uncertain data collection methods and poor-quality or malfunctioning data collection tools can produce unreliable crime data inquiries. Some traits are also more difficult to accurately quantify. To avoid this complexity, reliability estimation is used. It can measure how consistently a person is involved in the crime as a sort of average of the correlations between committing and silence, ranging from 0.0 to 1.0. Supervised machine learning is necessary for unlabeled clustering. When a crime is identified, clustering changes the classification [2]. Reliability is the application of crime data analytics, including AI machine learning, to predict when a committed crime investigation will fail or otherwise deteriorate so that it can be an inquiry or replaced before failing [4]. The software reliability growth model, divided into concave and S-shaped types shown in Figure 1, exhibits similar behavior, with the fault detection rate decreasing as faults are detected.

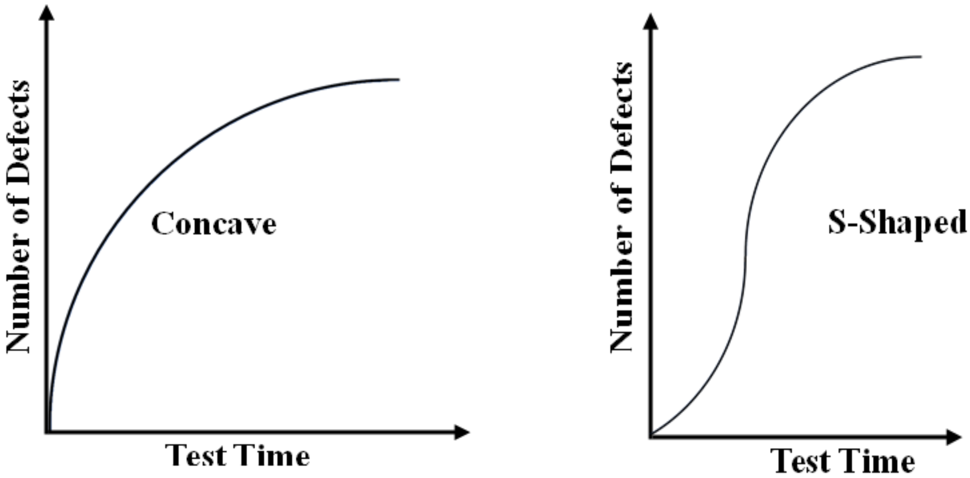


Figure 1. Concave and S-shaped models.

Defect density is the process of detecting defects in a crime application system during testing. It helps determine if a software system is ready for release, as proposed by Pushpa in 2019 [5]. However, identifying complete defects is challenging, especially for high-reliability software. To estimate defects, exponential software test coverage is used to measure thoroughness and estimate residual defect density. This method is easier to understand and visually observe. Reliability models are then used to evaluate the results.

The following six sections make up the correlation in this essay: Section 1's introduction and the proposed work in Section 2 using neutrosophic logic, a non-homogeneous poisoning process for crime clusters are covered in Section 2.1. Reinforcement learning is used for crime data analysis in Section 2.2. Uncertain cluster crime using least squares estimation in Section 2.3. A discussion of the experimental result is included in Section 2.4. The summary and projections for the future are found in Section 3 of the paper. References make mention of Section 4.

2. Proposed Work

The study utilizes neutrosophic logic and the non-homogenous Poisson process to analyze an uncertain crime cluster, focusing on the impact of software reliability on system reliability. The Contributions of this work are:

- To use hyperparameter control in the machine learning process using reinforcement learning on the uncertain crime cluster for a concave shape.
- To improve the uncertain cluster of crime using software reliability growth models.

The crime department utilizes a machine learning-based method called neutrosophic logic and a non-homogeneous Poisson process for crime investigation, which has a time limit for clusters.

2.1 Non-homogeneous Poisson Process-based Neutrosophic Logic for Crime Clusters

Neutrosophic logic is being utilized to create a non-homogeneous Poisson process for crime clusters. Veeraraghavan [6] introduced the Poisson process in stochastic processes $\{N(t) | t>=0\}$, counting actions and time t, for analyzing the non-homogeneous Poisson process on neutrosophic logic cluster criminals. N(t) is a random variable influenced by $N(t_n)$, which represents the number of crime cases identified at a specific time t and the number of criminals at time t_n.

$$P[N(t) = j \mid N(t_n) = i] = P[N(t) - N(t_n)] = j - i \quad (1)$$

where P[N(t) =j| process ending time], P[N(t) =i| Process Starting Time], j-i represents the process execution time. The neutrosophic logic rule can be used for continuous time-based Poisson processes, where criminals are involved in every crime detection system by Miguel Melgarejo [7].

$$\int_0^t N(t)dt = \int_0^1 (\mathrm{T} + \mathrm{I} + \mathrm{F})dt \quad (2)$$

The neutrosophic logic variable values in the same function N (t, s) vary between 0 to 1, as shown by evaluating the stochastic process $\int N(t)dt = \int (0 \leq \mathrm{T} + \mathrm{I} + \mathrm{F} \leq 1)$dt. The three-time interval crime data clusters in Neutrosophic logic, containing Certainty (T), Uncertainty (I), and False (F) which is not a criminal, is chosen and taken in the Non-homogeneous Poisson process. The rate parameter may change over time, and the general rate purpose function is given as λ(t). Here, T, I, and F are standard or non-standard real subsets of]-0, 1+[with not certainly any fitting together between them by Florentine [8].

$$\lambda_{a,b} = = \int_a^b \lambda(t)dt \quad (3)$$

The number of $\lambda_{a,b}$ on sets in the time interval (a, b], represented as N(b) - N(a), follows a poison process with associated parameters.

$$\mathrm{P}[\mathrm{N}(b) - \mathrm{N}(a) = \mathrm{K}] = \frac{e^{-\lambda_{a,b}}(\lambda_{a,b})^K}{K!}, K = 0.1, .., n \quad (4)$$

where K is the no. of events in the time interval between (a, b).
A time reason purpose in a Non-homogeneous Poisson process can be deterministic or autonomous, similar to a Cox procedure when λ(t) equals a constant rate proposed by Prasad [4].

2.2 Reinforcement Learning Used for Crime Cluster Data Analysis

Reinforcement learning (RL) is a method for customizing hyperparameters in crime data, transforming it into a supervised learning problem for model training, starting with a crime state and predicting an inquiry or investigation action introduced by Jagan Mohan [4, 5]. To anticipate future crime incentives, the model uses a discretized grid of hyperparameters, an uncertainty of crime loss function, policy curves, and qualitative learning techniques H: r = M (H). If a Reinforcement Learning model R is used to predict a value q with H and r, then q = R (H, r). The following R square error is minimized by the model (where g represents the discount rate for future rewards): (q' - (r + g*max q)) ^2. The network uses a linear layer output to predict q, simplifying policy gradient management and functioning as a classifier.

Next reward (Agreed/Silent) =M (next H). The crime type model is optimal for Hyperparameters with high crime rewards and silent low reward Hyperparameters, addressing the multi-label classification problem by Zhu et al. [9]. Cross entropy can be utilized to enhance the probability of a model producing certain Hyperparameters to 1, indicating our preference for them. L= (next H | current H, current r) * log e^{-P} accomplishes precisely that, but also balances the sample and reward value: L is equal to (next reward) *log e^{-P} (next H | current H, current r), where 0<-P<1.

2.3 Uncertain Cluster Crime using Least Square Estimation

Non-homogeneous Poisson process-based neutrosophic logic is utilized in crime case investigation to estimate the uncertainty of criminal cluster data using small sample sizes by Farrell [10]. It estimates Hyperparameters using failure intensity and best-possible mean values, obtaining coefficients for the equation Y= a + bX. The text discusses the use of Least Square estimation to estimate the probability of an uncertain cluster crime by Tsao Min [11].

Regression equation of x on y:

$$\sum x = b\sum y + Na \tag{5}$$

$$\sum xy = b\sum y^2 + a\sum y \tag{6}$$

Regression equation of y on x:

$$\sum y = b\sum x + Na \tag{7}$$

$$\sum xy = b\sum x^2 + a\sum x \tag{8}$$

The values of a and b can be easily determined by calculating the normal formula, allowing for easy determination of y and x.

The analysis of regression equations requires determining the appropriate criminal for the study. Establishing the relationship between dependent and independent criminals is crucial. Correlation, the linear relationship between two crime victims, is essential for this study, measured between observed variables by Win Bernic [12].

$$r = \frac{\sum_{i=1}^{n} y_i(x_i - \bar{x})}{\sqrt{[\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2]}} \tag{9}$$

The regression model uses a coefficient r to represent the mean of observed criminals, with values ranging from -1 to 1. A positive relationship indicates an increase or decrease in both criminals simultaneously, while a zero result indicates no or small linear relationship. A good fit includes a highly correlated dependent variable and independent criminals by Prasanth Sharma [13].

Independent criminals in regression can cause non-generalized, overfit models, leading to multicollinearity and conditioned X^TX. Perfect linear dependence can cause singular X^TX and infinite least squares estimates. The validity of a regression model is ensured by studying the residual standard error.

$$RSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}{n-k}} \tag{10}$$

The equation estimates the difference between fitted and observed values, aiding in model cross-validation to prevent overfitting, and is explained in a separate section by Prasanth Sharma [14].

2.4 Uncertain Crime Cluster Using Logistic Regression

Logistic regression is a popular machine learning algorithm used to predict categorical dependent variables using independent variables. It uses a "Concave" shaped logistic function to predict probabilistic values between 0 and 1, similar to Linear Regression. This technique is used for classification problems, rather than regression uncertainty problem, and is similar to Linear Regression in its approach. Logistic Regression is a crucial machine learning algorithm that provides probabilities and classifies data using continuous and discrete datasets. It helps identify the most effective variables for classification in criminal investigations by Prasanth Sharma [14].

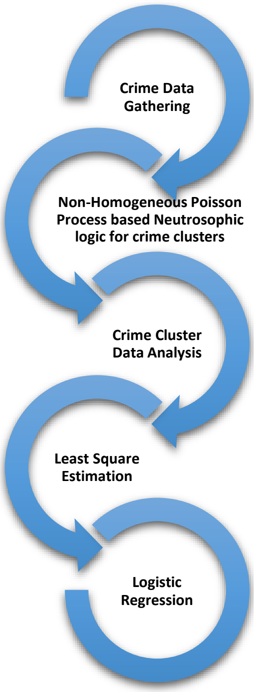


Figure 2. Working process to analyze an uncertain crime.

The categories of uncertain criminal data used include Murder, Rape, Robbery, and Auto-Theft as shown in Figure 3.

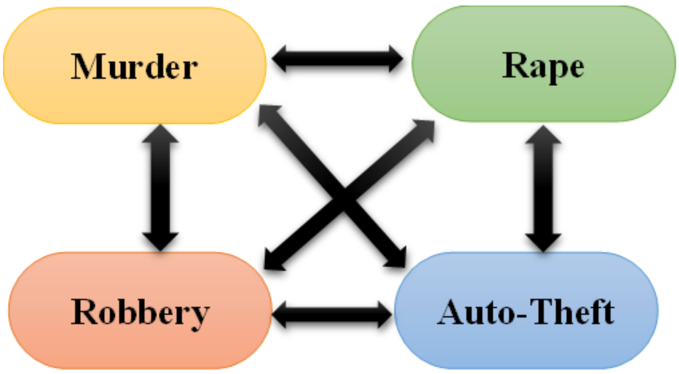
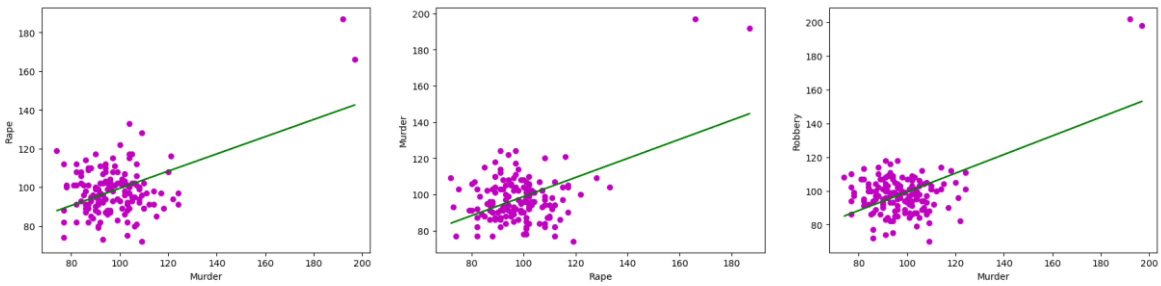


Figure 3. Regression statistics of uncertain criminals.

It will produce 12 combinations of regression analysis for each one that will be shown below in Figure 4 (a to k):



(a) (b) (c)

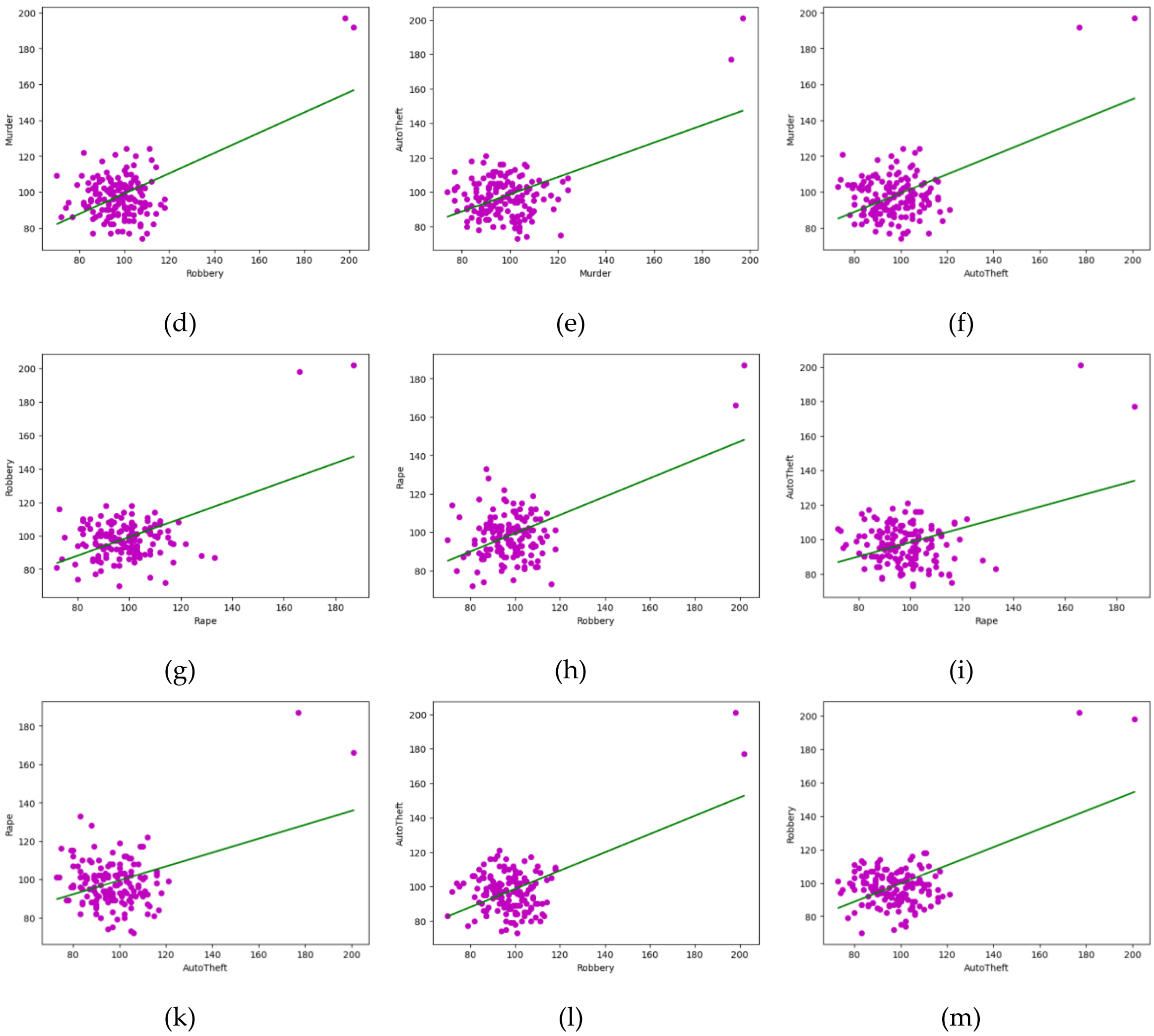


Figure 4. Regression analysis on uncertain criminals data.

3. Experimental Results

Experimental results in Tables (A and 1) show reliability estimation of criminal cases using neutrosophic logic and logistic regression on uncertain crime clusters, using indeterministic punishment data in a Concave-shape figure as shown in Figures 5-7.

Table 1. Crime punishment of uncertain criminals.

Regression Statistics	
Multiple R	0.668981671
R Square	0.447536476
Adjusted R Square	0.443561918
Standard Error	0.082019713
Observations	141

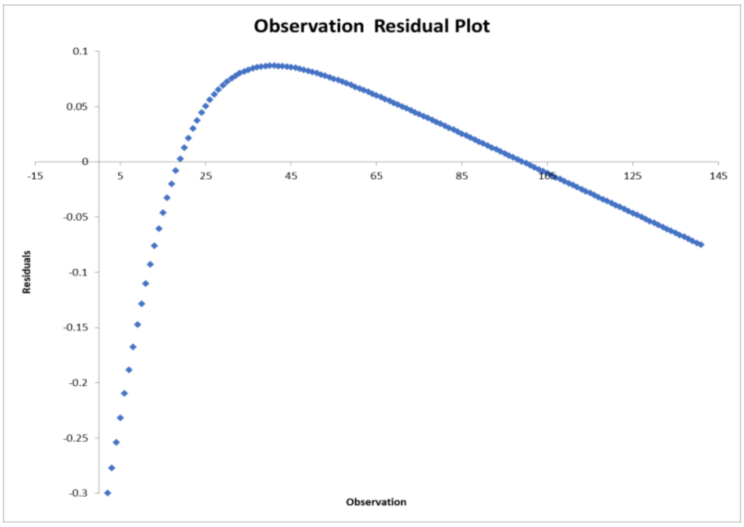


Figure 5. Observation residuals for uncertain criminal's punishment of crime cases.

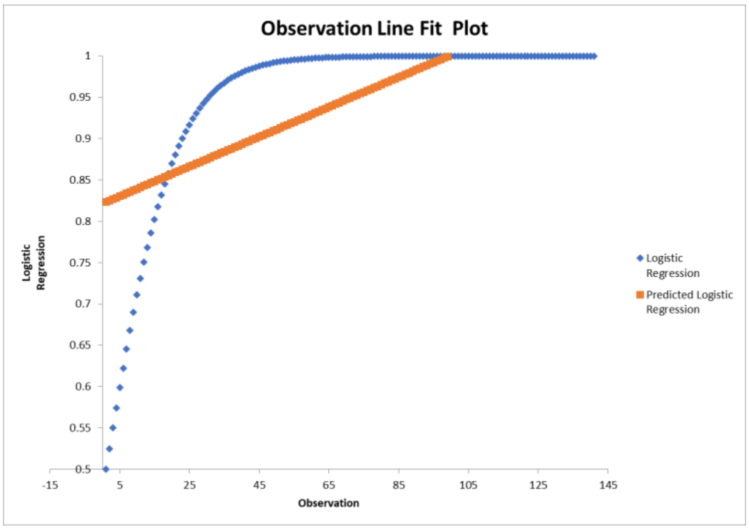


Figure 6. Observation line fit for uncertain criminal's punishment of crime cases.

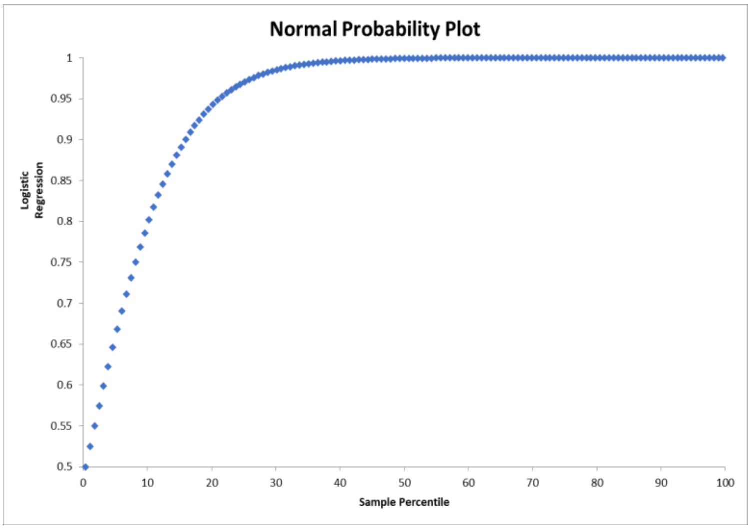


Figure 7. Concave-Shape of uncertain criminal's punishment of crime cases.

The software reliability growth model concave indicates a decrease in detection rate as faults are identified in crimes.

4. Conclusion

The likelihood that criminal data will work even if an investigation fails in a certain context has a big impact on cluster reliability. The study's main objective was to estimate software reliability models for a hazy crime cluster. In this respect, the criminal cluster predicts the non-homogeneous Poisson process, neutrosophic logic, and reinforcement learning technique. Using non-homogeneous Poisson process crime cluster data, logistic and least squares regression estimation, and neutrosophic logic-based crime cluster data, reinforcement learning classifies crimes, making it easier to anticipate crime probability based on crime data studied.

Acknowledgments
The author is grateful to the editorial and reviewers, as well as the correspondent author, who offered assistance in the form of advice, assessment, and checking during the study period.
Author Contributions
All authors contributed equally to this research.
Data availability
The datasets generated during and/or analyzed during the current study are not publicly available due to the privacy-preserving nature of the data but are available from the corresponding author upon reasonable request.
Funding
This research was not supported by any funding agency or institute.
Conflict of interest
The authors declare that there is no conflict of interest in the research.

References

1. Pramanik, A., Das, A. K., & Ding, W. (2023). Graph based fuzzy clustering algorithm for crime report labelling. Applied Soft Computing, 141, 110261. https://doi.org/10.1016/j.asoc.2023.110261
2. Vaidya, O., Mitra, S., Kumbhar, R., Chavan, S., & Patil, M. R. (2018). Crime rate prediction using data clustering algorithms. International Research Journal of Engineering and Technology (IRJET), 2395-0056.
3. Adeyiga, J. A., Olabiyisi, S. O., & Omidiora, E. O. (2020). A comparative analysis of selected clustering algorithms for criminal profiling. Nigerian Journal of Technology, 39(2), 464-471. https://doi.org/10.4314/njt.v39i2.16
4. Prasad, R. S., Supriya, N., & Mohan, G. K. (2011). Detection of Reliable Software Using SPRT. International Journal of Advanced Computer Science and Applications, 2(8).
5. Pushpa Latha Mamidi, R.Subba Rao, and R.N.V.Jagan Mohan (2019). Software Reliability Growth Model (SRGM) Based on Inverse Half-Logistic Distribution, International Journal of Recent Technology and Engineering (IJRTE), 8.
6. Veeraraghavan, M. (2004). Stochastic processes. Technical Report.
7. Melgarejo, M., Rodriguez, C., Mayorga, D., & Obregón, N. (2019). Time Series from Clustering: An Approach to Forecast Crime Patterns. In Recent Trends in Artificial Neural Networks-from Training to Prediction. IntechOpen. https://doi.org/10.5772/intechopen.89561
8. Smarandache, F. (2010). Neutrosophic logic-a generalization of the intuitionistic fuzzy logic. Multispace & multistructure. Neutrosophic transdisciplinarity (100 collected papers of science), 4, 396.
9. Zhu, X., Xu, J., Ge, J., Wang, Y., & Xie, Z. (2023). Multi-task multi-agent reinforcement learning for real-time scheduling of a dual-resource flexible job shop with robots. Processes, 11(1), 267. https://doi.org/10.3390/pr11010267

10. Farrell, G., & Pease, K. (2018). Prediction and Crime Clusters, Encyclopedia of Criminology and Criminal Justice, pp 3862–3871.
11. Tsao, M. (2023). Group least squares regression for linear models with strongly correlated predictor variables. Annals of the Institute of Statistical Mathematics, 75(2), 233-250. https://doi.org/10.1007/s10463-022-00841-7.
12. Wim Bernice et al (2023): A framework for estimating crime location choice based on awareness space, Crime Science, BMC, 9.
13. Prashant S. (2023). Different types of regression models, Analytics Vidhya.

Appendix

Table A. Regression statistics of uncertain criminals.

RESIDUAL OUTPUT				PROBABILITY OUTPUT	
Observation	Predicted Logistic Regression	Residuals	Standard Residuals	Percentile	Logistic Regression
1	0.822998174	-0.322998174	-3.95219566	0.354609929	0.5
2	0.824798955	-0.299819768	-3.668585398	1.063829787	0.524979187
3	0.826599736	-0.276765739	-3.386497016	1.773049645	0.549833997
4	0.828400517	-0.253958001	-3.107422236	2.482269504	0.574442517
5	0.830201299	-0.231513638	-2.832793715	3.191489362	0.59868766
6	0.83200208	-0.209542749	-2.563958586	3.90070922	0.622459331
7	0.833802861	-0.188146555	-2.30215542	4.609929078	0.645656306
8	0.835603642	-0.16741587	-2.048495403	5.319148936	0.668187772
9	0.837404423	-0.147429942	-1.803948209	6.028368794	0.689974481
10	0.839205204	-0.128255702	-1.569332797	6.737588652	0.710949503
11	0.841005986	-0.109947407	-1.345313068	7.446808511	0.731058579
12	0.842806767	-0.092546661	-1.132398081	8.156028369	0.750260106
13	0.844607548	-0.076082764	-0.93094635	8.865248227	0.768524783
14	0.846408329	-0.060573346	-0.741173587	9.574468085	0.785834983
15	0.84820911	-0.046025222	-0.563163187	10.28368794	0.802183889
16	0.850009891	-0.032435415	-0.396878736	10.9929078	0.817574476
17	0.851810673	-0.019792287	-0.242177816	11.70212766	0.832018385
18	0.853611454	-0.008076719	-0.098826481	12.41134752	0.845534735
19	0.855412235	0.0027367	0.03348618	13.12056738	0.858148935
20	0.857213016	0.01267851	0.155133852	13.82978723	0.869891526
21	0.859013797	0.021783281	0.26653955	14.53900709	0.880797078
22	0.860814578	0.0300886	0.368163184	15.24822695	0.890903179
23	0.862615359	0.037634151	0.460490312	15.95744681	0.900249511
24	0.864416141	0.044460898	0.544022176	16.66666667	0.908877039
25	0.866216922	0.050610382	0.619267064	17.37588652	0.916827304
26	0.868017703	0.056124117	0.686732959	18.08510638	0.92414182
27	0.869818484	0.061043096	0.746921428	18.79432624	0.93086158
28	0.871619265	0.065407379	0.800322661	19.5035461	0.937026644
29	0.873420046	0.069255778	0.847411552	20.21276596	0.942675824

30	0.875220828	0.072625609	0.888644708	20.92198582	0.947846437
31	0.877021609	0.075552518	0.924458273	21.63120567	0.952574127
32	0.87882239	0.078070355	0.955266449	22.34042553	0.956892745
33	0.880623171	0.080211106	0.981460612	23.04964539	0.960834277
34	0.882423952	0.082004858	1.00340891	23.75886525	0.964428811
35	0.884224733	0.083479802	1.021456272	24.46808511	0.967704535
36	0.886025515	0.084662255	1.035924728	25.17730496	0.970687769
37	0.887826296	0.085576711	1.047113983	25.88652482	0.973403006
38	0.889627077	0.086245902	1.055302184	26.59574468	0.975872979
39	0.891427858	0.086690871	1.060746815	27.30496454	0.978118729
40	0.893228639	0.086931055	1.063685699	28.0141844	0.980159694
41	0.89502942	0.08698437	1.064338055	28.72340426	0.98201379
42	0.896830202	0.086867299	1.062905583	29.43262411	0.983697501
43	0.898630983	0.086594986	1.059573564	30.14184397	0.985225968
44	0.900431764	0.086181318	1.054511943	30.85106383	0.986613082
45	0.902232545	0.08563902	1.047876398	31.56028369	0.987871565
46	0.904033326	0.084979731	1.039809361	32.26950355	0.989013057
47	0.905834107	0.084214091	1.030441008	32.9787234	0.990048198
48	0.907634889	0.083351813	1.019890202	33.68794326	0.990986701
49	0.90943567	0.082401759	1.008265375	34.39716312	0.991837429
50	0.911236451	0.081372008	0.995665369	35.10638298	0.992608459
51	0.913037232	0.080269917	0.982180221	35.81560284	0.993307149
52	0.914838013	0.079102185	0.967891892	36.5248227	0.993940199
53	0.916638794	0.077874907	0.952874949	37.23404255	0.994513701
54	0.918439576	0.076593623	0.937197199	37.94326241	0.995033198
55	0.920240357	0.07526337	0.920920268	38.65248227	0.995503727
56	0.922041138	0.073888724	0.904100146	39.36170213	0.995929862
57	0.923841919	0.072473841	0.886787677	40.07092199	0.99631576
58	0.9256427	0.071022493	0.869029021	40.78014184	0.996665193
59	0.927443481	0.069538102	0.850866068	41.4893617	0.996981584
60	0.929244262	0.068023777	0.832336827	42.19858156	0.997268039
61	0.931045044	0.066482333	0.813475771	42.90780142	0.997527377
62	0.932845825	0.064916327	0.794314164	43.61702128	0.997762151
63	0.934646606	0.063328074	0.774880349	44.32624113	0.99797468
64	0.936447387	0.061719674	0.75520002	45.03546099	0.998167061
65	0.938248168	0.060093031	0.735296463	45.74468085	0.998341199
66	0.940048949	0.058449868	0.715190779	46.45390071	0.998498818
67	0.941849731	0.056791749	0.694902088	47.16312057	0.99864148
68	0.943650512	0.05512009	0.67444771	47.87234043	0.998770601
69	0.945451293	0.053436171	0.653843335	48.58156028	0.998887464
70	0.947252074	0.051741155	0.633103172	49.29078014	0.998993229
71	0.949052855	0.050036094	0.612240092	50	0.999088949
72	0.950853636	0.048321939	0.591265749	50.70921986	0.999175575
73	0.952654418	0.046599554	0.570190695	51.41843972	0.999253971

74	0.954455199	0.044869719	0.549024487	52.12765957	0.999324917
75	0.95625598	0.043133141	0.527775775	52.83687943	0.999389121
76	0.958056761	0.04139046	0.506452391	53.54609929	0.999447221
77	0.959857542	0.039642257	0.485061425	54.25531915	0.999499799
78	0.961658323	0.037889054	0.463609297	54.96453901	0.999547378
79	0.963459105	0.036131328	0.442101814	55.67375887	0.999590433
80	0.965259886	0.034369508	0.420544238	56.38297872	0.999629394
81	0.967060667	0.032603983	0.398941327	57.09219858	0.99966465
82	0.968861448	0.030835105	0.377297389	57.80141844	0.999696553
83	0.970662229	0.029063193	0.355616325	58.5106383	0.999725422
84	0.97246301	0.027288535	0.333901664	59.21985816	0.999751545
85	0.974263792	0.025511392	0.312156599	59.92907801	0.999775183
86	0.976064573	0.023732	0.290384021	60.63829787	0.999796573
87	0.977865354	0.021950574	0.268586547	61.34751773	0.999815928
88	0.979666135	0.020167307	0.246766543	62.05673759	0.999833442
89	0.981466916	0.018382373	0.224926153	62.76595745	0.99984929
90	0.983267697	0.016595932	0.203067314	63.4751773	0.99986363
91	0.985068479	0.014808127	0.181191782	64.18439716	0.999876605
92	0.98686926	0.013019087	0.159301145	64.89361702	0.999888347
93	0.988670041	0.01122893	0.137396839	65.60283688	0.999898971
94	0.990470822	0.009437762	0.115480165	66.31205674	0.999908584
95	0.992271603	0.00764568	0.093552299	67.0212766	0.999917283
96	0.994072384	0.005852769	0.071614306	67.73049645	0.999925154
97	0.995873165	0.00405911	0.049667149	68.43971631	0.999932276
98	0.997673947	0.002264774	0.0277117	69.14893617	0.99993872
99	0.999474728	0.000469824	0.005748748	69.85815603	0.999944551
100	1.001275509	-0.001325681	-0.016220994	70.56737589	0.999949828
101	1.00307629	-0.003121688	-0.038196878	71.27659574	0.999954602
102	1.004877071	-0.004918149	-0.060178321	71.9858156	0.999958922
103	1.006677852	-0.006715021	-0.082164794	72.69503546	0.999962831
104	1.008478634	-0.008512266	-0.104155818	73.40425532	0.999966368
105	1.010279415	-0.010309846	-0.126150961	74.11347518	0.999969568
106	1.012080196	-0.012107732	-0.14814983	74.82269504	0.999972464
107	1.013880977	-0.013905893	-0.17015207	75.53191489	0.999975085
108	1.015681758	-0.015704303	-0.192157362	76.24113475	0.999977456
109	1.017482539	-0.017502939	-0.214165414	76.95035461	0.999979601
110	1.019283321	-0.019301779	-0.236175964	77.65957447	0.999981542
111	1.021084102	-0.021100803	-0.258188775	78.36879433	0.999983299
112	1.022884883	-0.022899995	-0.280203631	79.07801418	0.999984888
113	1.024685664	-0.024699338	-0.302220337	79.78723404	0.999986326
114	1.026486445	-0.026498818	-0.324238717	80.4964539	0.999987627
115	1.028287226	-0.028298422	-0.346258613	81.20567376	0.999988805
116	1.030088008	-0.030098138	-0.36827988	81.91489362	0.99998987
117	1.031888789	-0.031897955	-0.390302387	82.62411348	0.999990834

118	1.03368957	-0.033697864	-0.412326017	83.33333333	0.999991706
119	1.035490351	-0.035497856	-0.434350662	84.04255319	0.999992495
120	1.037291132	-0.037297923	-0.456376227	84.75177305	0.99999321
121	1.039091913	-0.039098058	-0.478402622	85.46099291	0.999993856
122	1.040892695	-0.040898254	-0.500429771	86.17021277	0.999994441
123	1.042693476	-0.042698506	-0.5224576	86.87943262	0.99999497
124	1.044494257	-0.044498809	-0.544486045	87.58865248	0.999995448
125	1.046295038	-0.046299157	-0.566515048	88.29787234	0.999995881
126	1.048095819	-0.048099546	-0.588544554	89.0070922	0.999996273
127	1.0498966	-0.049899972	-0.610574518	89.71631206	0.999996628
128	1.051697382	-0.051700433	-0.632604894	90.42553191	0.999996949
129	1.053498163	-0.053500923	-0.654635644	91.13475177	0.999997239
130	1.055298944	-0.055301442	-0.676666732	91.84397163	0.999997502
131	1.057099725	-0.057101985	-0.698698126	92.55319149	0.99999774
132	1.058900506	-0.058902551	-0.720729796	93.26241135	0.999997955
133	1.060701287	-0.060703138	-0.742761717	93.97163121	0.999998149
134	1.062502069	-0.062503743	-0.764793865	94.68085106	0.999998326
135	1.06430285	-0.064304365	-0.786826218	95.39007092	0.999998485
136	1.066103631	-0.066105002	-0.808858756	96.09929078	0.999998629
137	1.067904412	-0.067905652	-0.830891462	96.80851064	0.99999876
138	1.069705193	-0.069706316	-0.852924321	97.5177305	0.999998878
139	1.071505974	-0.07150699	-0.874957316	98.22695035	0.999998984
140	1.073306755	-0.073307674	-0.896990436	98.93617021	0.999999081
141	1.075107537	-0.075108368	-0.919023669	99.64539007	0.999999168

Received: 26 Dec 2023, **Revised:** 28 Mar 2024,
Accepted: 29 Apr 2024, **Available online:** 02 May 2024.