SJIF Impact Factor (2025): 8.688| ISI I.F. Value: 1.241| Journal DOI: 10.36713/epra2016 ISSN: 2455-7838(Online)
EPRA International Journal of Research and Development (IJRD)
Volume: 10 | Issue: 7 | July 2025 - Peer Reviewed Journal

CRIME TYPE AND OCCURRENCE PREDICTION USING MACHINE LEARNING

Dr.V.Vijayakumar[1], S.Kottish.M.E.[2]
[1]Head of the Department, Department of Computer Science and Engineering, AVS Engineering College, Salem.
[2]Research Scholar, Department of Computer Science and Engineering, AVS Engineering College, Salem.

ABSTRACT
The ability to analyze this amount of data with its inherent complexities without using computational support puts a strain on human resources. This paper examines the current techniques that are used to predict crime and criminality. Over time, these techniques have been refined and have achieved limited success. data Mining is the procedure which includes evaluating and examining large pre-existing databases in order to generate new information which may be essential to the organization. The criminals can also be predicted based on the crime data. The main aim of this work is to perform a survey on the supervised learning and unsupervised learning techniques that has been applied towards criminal identification. This paper presents the survey on the crime analysis and crime prediction using several data Mining techniques. To inform that to perform ensure a crime free society and to identify crime characteristics.

CHAPTER 1
INTRODUCTION

The role of computers has been increased in all walks of life from the finance sector to supermarkets. In recent years police forces have been enhancing their Traditional method of crime reporting with new technological advancements to increase their output by efficiently recording crimes to aid their investigation. Data is not just a record of crimes, it also contains valuable information. Crime analysis is a process which includes exploring the behavior of the crimes, detecting crimes and their relationships with criminals. The huge volume of crime and criminal datasets and the complexity of relationships between these kinds of information have made criminology an appropriate field for applying data mining techniques.

CHAPTER 2
LITERATURE SURVEY

XINGCHEN YU, "DESIGN OF CROSS-BORDER NERWORK CRIME DETECTION SYSTEM BASED ON PSE AND BIG DATA ANALYSIS",2020.

In order to speed up the detection of cybercrime worldwide, a new cross-border cybercrime detection system is designed by introducing the PSE theory and the principle of big data analysis. In the TFTP server, the U-boot network development board and Open Stack crime information detection component are connected to build the hardware running environment of the cross-border network crime detection system. On this basis, through the analysis of the characteristics of detection information, the calculation of cross-border network detection domain and the directional planning of network crime information, the software operating environment of the detection system is built, and the cross border network crime detection system based on PSE and big data analysis is designed with the basis of hardware implementation. The comparative experimental results show that, compared with the conventional case detection situation, the average detection time of criminal cases is basically maintained between 3-5 days after the application border cyber crime detection system, and the accuracy of criminal location is maintained at more than 90%. Supported by three basic processes: TFTP server design, U-boot network development board design and Open Stack criminal information detection component connection, the hardware operation environment of cross-border network crime detection system based on PSE and big data analysis is built.

CHAPTER 3
SYSTEM SPECIFICATION

3.1. HARDWARE REQUIREMENTS

- 1 GB RAM
- 80 GB Hard Disk
- Above 2GHz Processor
- Windows OS

3.2. SOFTWARE REQUIREMENTS

- Windows 7 OS 32 bit
- PYTHON
- ANNANCONDA

SJIF Impact Factor (2025): 8.688| ISI I.F. Value: 1.241| Journal DOI: 10.36713/epra2016 ISSN: 2455-7838(Online)
EPRA International Journal of Research and Development (IJRD)
Volume: 10 | Issue: 7 | July 2025 - Peer Reviewed Journal

CHAPTER 4
SYSTEM STUDY
4.1. EXISTING SYSTEM
DESCRIPTION

Integrates and reduces the extracted crime data into structured 5,038 crime instances. We represent these instances using 35 predefined crime attributes. Safeguard measures are taken for the crime database accessibility. Rest four modules are useful for crime detection, criminal identification and prediction, and crime verification, respectively. Payload attribution systems (PAS) are one of the most important tools of network forensics for detecting an offender after the occurrence of a cyber crime. A PAS stores the network traffic history in order to detect the source and destination pair of a certain data stream in case a malicious activity occurs on the network. Our approach contributes in the betterment of the society by helping the investigating agencies in crime detection and criminals' identification, and thus reducing the crime rates.

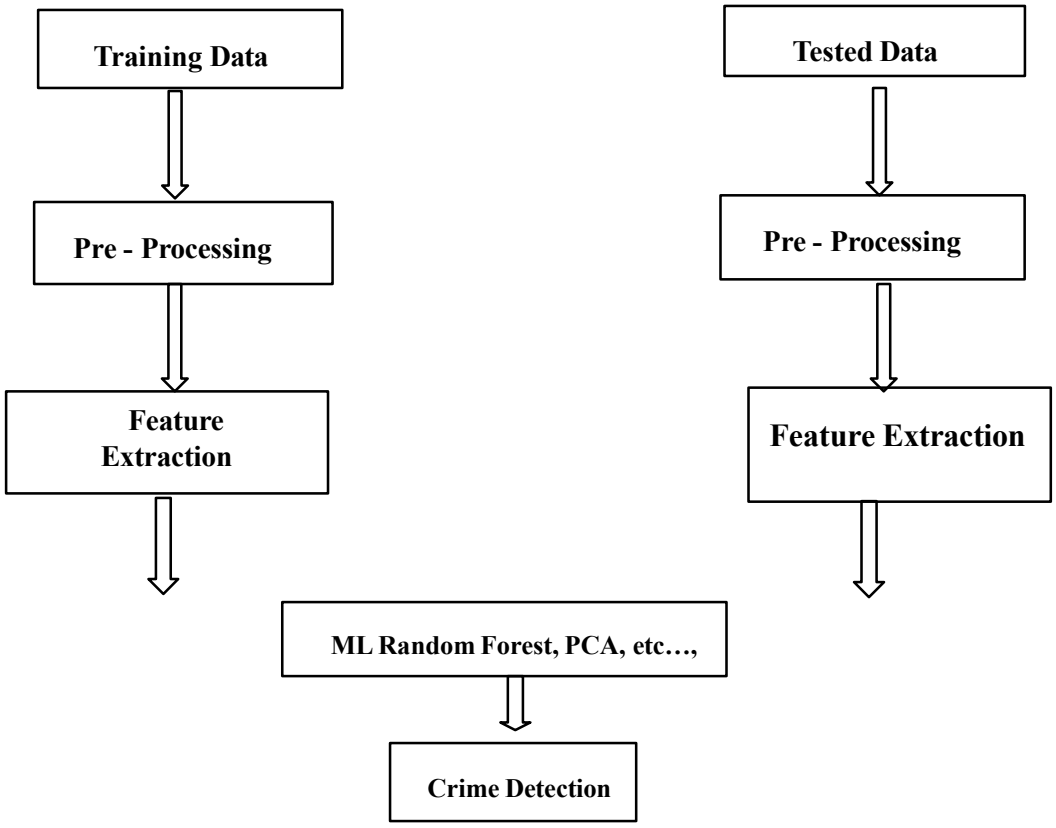
DISADVANTAGES

- Reduction in accuracy
- Miss classification
- Less prediction

4.2. PROPOSED SYSTEM
FEATURES

In his project, the objective would be to train a model for prediction. The training would be done using the training data set which will be validated using the test dataset. Building the model will be done using better algorithm depending upon the accuracy. The Random forest algorithm classification and other algorithm will be used for crime prediction. Visualization of dataset is done to analyze the crimes which may have occurred in the country. This work helps the law enforcement agencies to predict and detect crimes. An improve with an enhanced feature selection method and attribute-importance factor is applied to produce a better and faster Random forest algorithm based on the information entropy which is explicitly derived from a series of training data sets from several classes.

4.2.1BLOCK DIAGRAM



SJIF Impact Factor (2025): 8.688| ISI I.F. Value: 1.241| Journal DOI: 10.36713/epra2016 ISSN: 2455-7838(Online)
EPRA International Journal of Research and Development (IJRD)
Volume: 10 | Issue: 7 | July 2025 - Peer Reviewed Journal

CHAPTER 5
SYSTEM DESIGN AND DEVELOPMENT
5.1.FILE DESIGN
Data are accumulated into files that are processed or maintained by the system. The systems analyst is responsible for designing files, determining their contents and selecting a method for organizing the data. Data Item Individual elements of data are called data items also known as fields or simply items. Record Key is to distinguish one specific record from another, systems analysts select one data item in the record that is likely to be unique in all records of a file and use it for identification purposes.

5.2.INPUT DESIGN
Input design is the raw data that is processed to produce output. The decisions made during the input designs are

- To provide cost effective method of input
- To achieve the highest possible of accuracy
- To ensure that the code is understand by the user

It is important to design appropriate data input methods to prevent errors while entering data. These methods depend on whether the data is entered by customers in forms manually and later entered by data entry operators, or data is directly entered by users

CHAPTER 6
TESTING AND IMPLEMENTATION
6.1. INTRODUCTION
The various levels of testings are

1. White Box Testing
2. Black Box Testing
3. Unit Testing
4. Functional Testing
5. Performance Testing
6. Integration Testing
7. Objective
8. Integration Testing
9. Validation Testing
10. System Testing
11. Structure Testing
12. Output Testing
13. User Acceptance Testing

6.2. TESTING OBJECTIVES
1.White Box Testing

White-box testing (also known as **clear box testing**, **glass box testing**, **transparent box testing**, and **structural testing**) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing). In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit, e.g. in-circuit testing (ICT).

White-box test design techniques include:

- Control flow testing
- Data flow testing
- Branch testing
- Path testing
- Statement coverage
- Decision coverage

2.Black Box Testing

Black-box testing is a method of software testing that examines the functionality of an application (e.g. what the software does) without peering into its internal structures or workings (see white-box testing). This method of test can be applied to virtually every level of software testing: unit, integration, system and acceptance. It typically comprises most if not all higher level testing, but can also dominate unit testing as well as white box testing.

SJIF Impact Factor (2025): 8.688| ISI I.F. Value: 1.241| Journal DOI: 10.36713/epra2016 ISSN: 2455-7838(Online)
EPRA International Journal of Research and Development (IJRD)
Volume: 10 | Issue: 7 | July 2025 - Peer Reviewed Journal

3.Unit Testing

In computer programming, **unit testing** is a method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested to determine if they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit could be an entire module, but is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are created by programmers or occasionally by white box testers during the development process.

4.Functional Testing

Functional testing is a quality assurance (QA) process and a type of black box testing that bases its test cases on the specifications of the software component under test. Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered (not like in white-box testing). Functional Testing usually describes what the system does.

5.Performance Testing

In software engineering, **performance testing** is in general testing performed to determine how a system performs in terms of responsiveness and stability under a particular workload. It can also serve to investigate, measure, validate or verify other quality attributes of the system, such as scalability, reliability and resource usage.

6.Integration Testing

Integration testing (sometimes called **integration and testing**, abbreviated **I&T**) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

7.Top-down and Bottom-up

Bottom Up Testing is an approach to integrated testing where the lowest level components are tested first, then used to facilitate the testing of higher level components. The process is repeated until the component at the top of the hierarchy is tested.

Top Down Testing is an approach to integrated testing where the top integrated modules are tested and the branch of the module is tested step by step until the end of the related module.

Sandwich Testing is an approach to combine top down testing with bottom up testing.

8.Verification and validation

Verification and Validation are independent procedures that are used together for checking that a product, service, or system meets requirements and specifications and that it full fills its intended purpose. These are critical components of a quality management system such as ISO 9000. The words "verification" and "validation" are sometimes preceded with "Independent" (or IV&V), indicating that the verification and validation is to be performed by a disinterested third party.

9.System Testing

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

10.Structure Testing

It is concerned with exercising the internal logic of a program and traversing particular execution paths.

11.Output Testing

Output of test cases compared with the expected results created during design of test cases.

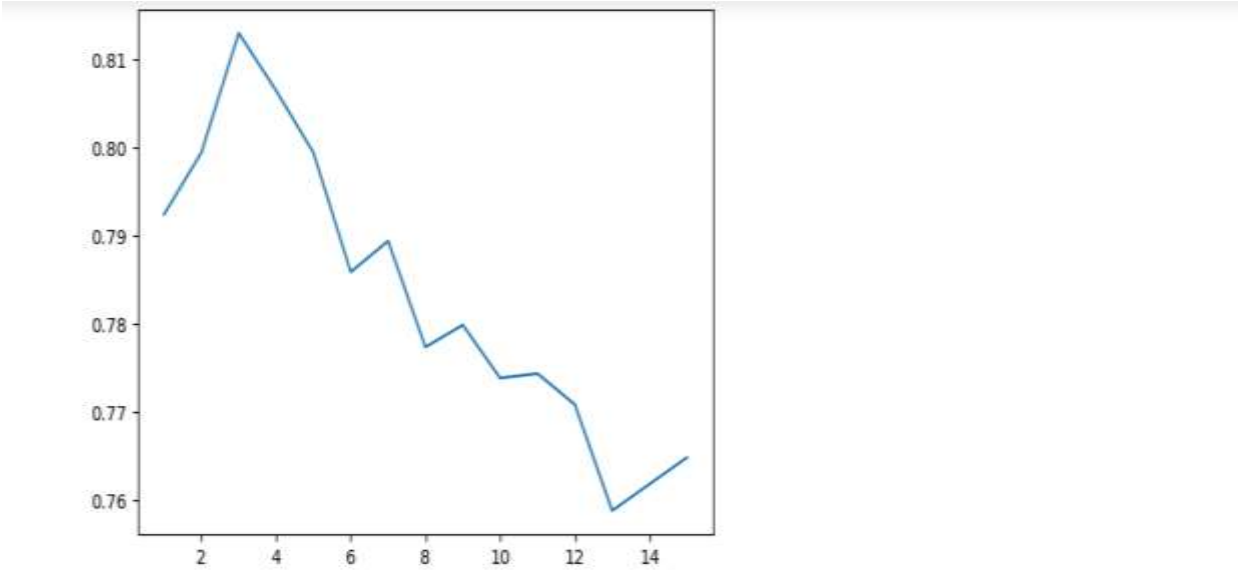
- Asking the user about the format required by them tests the output generated or displayed by the system under consideration.
- Here, the output format is considered into two was, one is on screen and another one is printed format.
- The output on the screen is found to be correct as the format was designed in the system design phase according to user needs.
- The output comes out as the specified requirements as the user's hard copy.

SJIF Impact Factor (2025): 8.688| ISI I.F. Value: 1.241| Journal DOI: 10.36713/epra2016 ISSN: 2455-7838(Online)
EPRA International Journal of Research and Development (IJRD)
Volume: 10 | Issue: 7 | July 2025 - Peer Reviewed Journal

CHAPTER 7
CONCLUSION

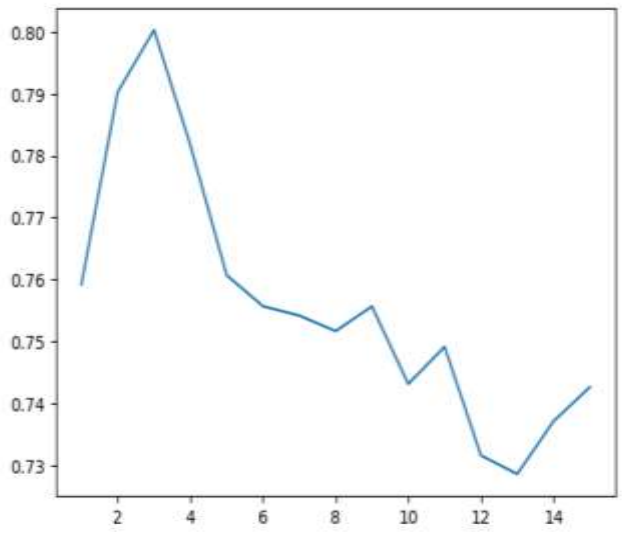
We have studied some known approaches for crime analysis and prediction concerned with data mining. Although many papers have been studied, only those papers with background in the crime prediction and criminal identification papers are compared with a theoretical study. Each paper has their own advantages and disadvantages. Each paper has its own individual approach for solving the crimes and criminal prediction. This is a theoretical study for several methods in identification of crime by the random forest algorithm gives better accuracy.

CHAPTER 8
SCREEN SHOT



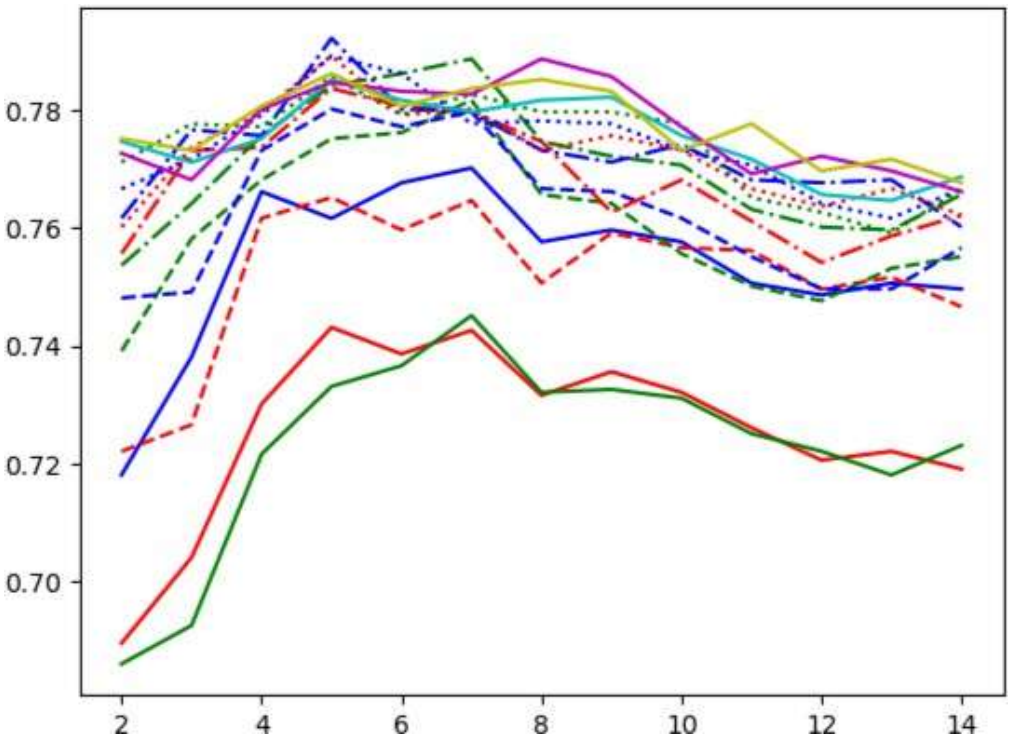

```
y is  [0.7923703731691037, 0.7993879170287527, 0.8129419024949309, 0.8064142768982758, 0.7993853981687887, 0.7858503041523
406, 0.7893653732320751, 0.7773227037442852, 0.7798377854183196, 0.7738227478243347, 0.7743265198171307, 0.770819007317288
3, 0.7587801161194444, 0.7617939320663467, 0.7648039697233032]
```

X is Crime rate




```
[0.759190690293573, 0.790293573128802, 0.8003161169254795, 0.7817294492512689, 0.7606919308321054, 0.7556680646339466, 0.
7541668240954144, 0.75165174242138, 0.7556479137542349, 0.7431065099935769, 0.7491240664475258, 0.7315688719285651, 0.7285
512776917167, 0.737078878099772, 0.7426027380007809]
```

SJIF Impact Factor (2025): 8.688| ISI I.F. Value: 1.241| Journal DOI: 10.36713/epra2016 ISSN: 2455-7838(Online)
EPRA International Journal of Research and Development (IJRD)
Volume: 10 | Issue: 7 | July 2025 - Peer Reviewed Journal



`10 PCA feature components range

CHAPTER 9
REFERENCES

1. *C. N. Satoshi Sekine, Kiyoshi Sudo, "Extended named entity hierarchy," in Third International Conference on Language Resources and Evaluation (LREC- 2002), February 2002, pp. 1818–1824.*
2. *G. Weir and N. Anagnostou, "Exploring newspapers: A case study in corpus analysis," in ICTATLL Workshop, August 2007.*
3. *I. C. H. Ku and G. Leroy, "Natural language processing and e-government: Crime information extraction from heterogeneous data sources," in Ninth International Conference on Digital Government Research, May 2008, pp. 162– 170.*
4. *S. Brin, "Extracting patterns and relations from the world wide web," in Selected Papers from the International Workshop on The World Wide Web and Databases, 1999, pp. 172–183.*
5. *E. Agichtein and L. Gravano, "Snowball: Extracting relations from large plain-text collections," in Proceedings of the Fifth ACM Conference on Digital Libraries, 2000, pp. 85–94.*
6. *Carlson, J. Betteridge, R. C. Wang, E. R. Hruschka, Jr., and T. M. Mitchell, "Coupled semi-supervised learning for information extraction," in Proceedings of the Third ACM International Conference on Web Search and Data Mining, 2010, pp. 101–110.*
7. *D. S. Batista, B. Martins, and M. J. Silva, "Semi-supervised bootstrapping of relationship extractors with distributional semantics," in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, September 2015, pp. 499–504.*
8. *C. Zhang, W. Xu, Z. Ma, S. Gao, Q. Li, and J. Guo, "Construction of semantic bootstrapping models for relation extraction," Knowledge-Based Systems, vol. 83, pp. 128 – 137, 2015.*
9. *T. Hasegawa, S. Sekine, and R. Grishman, "Discovering relations among named entities from large corpora," in Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics, no. 415, 2004.*