

# Prediction of DOS Attacks using Machine Learning Algorithms

**A Saritha<sup>\*1</sup>, B Rama Subba Reddy<sup>2</sup>, A Suresh Babu<sup>3</sup>**

<sup>1</sup>Research Scholar, Department of Computer Science Engineering, JNTUA College of Engineering, Anantapuramu, Andhra Pradesh, India

<sup>2</sup>Professor, Department of Computer Science Engineering, SV College of Engineering, Tirupati, Andhra Pradesh, India

<sup>3</sup>Professor and HOD, Department of Computer Science Engineering, JNTUA College of Engineering, Anantapuramu, Andhra Pradesh, India

**Email:** \*sarithaanchuri@gmail.com

**DOI:**

## Abstract

Unique-Intrusion identification framework assumes in fundamental job in recognizing both dynamic and inactive assaults and illicit system get to. There are such a significant number of calculations for recognizing system assaults utilizing information mining and Artificial Intelligence strategies. So, far the current systems were creating less precision and tedious. So, as to improve the precision and to decrease time taken, we are going to join both directed and unsupervised strategies. In this, we first use includes determination and the loads are connected in the proposed classifier, in which Naïve Bayes classifier is utilized. The objective of any probabilistic classifier is, with highlights  $x_0$  through  $x_n$  and classes  $c_0$  through  $c_k$ , to decide the likelihood of the highlights happening in each class, and to restore the doubtlessly class. In this way, for each class, we need to have the capacity to figure  $P(c_i/x_0, \dots, x_n)$ . The test results dependent on the KDD dataset demonstrate that the proposed technique not just performs well on recognizing DoS, Probe and R2L assaults, it likewise has noteworthy improvement for distinguishing U2R assaults.

**Keywords:** AI, highlight determination, interruption identification, organize security, oversee learning, unsupervised learning

## INTRODUCTION

With the broad development of web and system utilization as of late, individuals were depending on the web to discuss. Alongside, the noteworthy extension in system assets, assaults against the system are likewise rising. The requirement for system security has turned out to be a standout amongst the most test issues for systems administration frameworks and data foundation because of progressively wide spread digital burglary, misrepresentation and misuse. Arrangement of instruments have been taken to ensure web frameworks, including setting up of firewall and hostile to infection. Despite the fact that these static protection components can give a dimension of security, progressively

unique instruments, for example, interruption location frameworks (IDSs) ought to likewise be used [1-4]. As an imperative job in system security, the principle favorable position of IDS is to screen arrange traffic without influencing the execution of the system. It gives ongoing assurance against inside assaults, outer assaults and mis operations. Masterminding interruption recognition frameworks in vast and complex PC systems can fundamentally improve the nature of system security the board.

There are many proposed half and half methodologies which are utilizing AI and information mining procedures in IDS to tackle identification issues and to boost their focal points, bolster vector machine,

conventional calculation, Bayesian, K-Nearest Neighbor(K-NN), and so forth [5–13]. The arrangement of AI strategies connected in the writing comprises an exceptionally little subset of what is possibly material for the interruption location, and strategies that consolidating diverse methods are demonstrating better outcomes [14–16]. In light of broadly held conviction, assault execution elements and marks show generous variety starting with one assault class then onto the next, bringing about constraint of AI based assault identification technique. These examines worked on KDD interruption recognition datasets, which is a benchmark to assess diverse procedures, demonstrates that for digital assaults in the types of DoS (denial of administration) and Probe (surveillance and testing), calculations referenced above dependably perform well.

In this work, we center on the administered learning technique Naïve Bayesian characterization and unsupervised learning strategy various leveled bunching, attempting to improve the execution of interruption classifier. So, as to accomplish this target, we present element weighting and administered learning strategy Naïve Bayesian order. We initially perform highlight choice and dole out loads as indicated by their job for grouping. We utilize the chose list of capabilities as the contribution to our proposed calculation, and the loads of highlights are connected when playing out the calculation. By doling out loads to highlights, the grouping procedure for common assault class become reasonable. At that point we present unsupervised learning Hierarchical bunching calculation in Naïve Bayesian classifier. We start by treating every datum point as a solitary group, i.e., in the event that there are X information focuses in our dataset, at that point we have X bunches. We at that point select a separation metric that estimates

the separation between two groups. For instance, we will utilize normal linkage which characterizes the separation between two groups to be the normal separation between information focuses in the primary bunch and information focuses in the second group.

The primary thought of this paper can be outlined as pursues:

We proposed highlight choice and weighting technique dependent on the component pertinence investigation, in which data gain is determined for each element. We select the most pertinent highlights for interruption location and allot distinctive loads to these highlights, in order to make the characterization simpler.

We plan an unsupervised learning and directed learning joined calculation for assault grouping. The various leveled bunching calculation is utilized for choosing neighbors of test models, attempting to improve the execution of classifier.

Experiments are directed to assess the proposed strategy. The outcomes demonstrate that the proposed calculation performs well for characterizing system assaults.

## LITERATURE REVIEW

### Machine Learning

AI is a use of man-made consciousness (AI) that gives frameworks the capacity to consequently take in and improve as a matter of fact without being expressly customized. AI centers around the advancement of PC programs that can get to information and use it learn for themselves. AI comes in a wide range of flavors, contingent upon the calculation and its destinations. We can partition AI calculations into three fundamental gatherings dependent on their motivation: Supervised learning and unsupervised learning.

Regulated learning as the name demonstrates a nearness of manager as instructor. Essentially regulated learning is a learning in which we educate or train the machine utilizing information which is all around named that implies a few information is now labeled with right answer. From that point onward, machine is furnished with new arrangement of examples (data) so regulated learning calculation examinations the preparation data (set of preparing precedents) and produces a right result from marked information.

Supervised learning is classified into two categories of algorithms:

**Classification:** A grouping issue is the point at which the yield variable is a class, for example, "Red" or "blue" or "malady" and "no ailment".

**Regression:** A relapse issue is the point at which the yield variable is a genuine esteem, for example, "dollars" or "weight".

Unsupervised learning is the preparation of machine utilizing data that is neither arranged nor marked and enabling the calculation to follow up on that data without direction. Here the errand of machine is to assemble unsorted data as indicated by similitude, examples and contrasts with no earlier preparing of information.

In contrast to administered adapting, no educator is given that implies no preparation will be given to the machine. In this manner machine is limited to locate the shrouded structure in unlabeled information by our-self.

Unsupervised learning characterized into two classes of calculations:

**Clustering:** A bunching issue is the place you need to find the natural groupings in the information, for example, gathering clients by buying conduct.

**Association:** An affiliation rule learning issue is the place you need to find decides that depict substantial bits of your information, for example, individuals that purchase X additionally will in general purchase Y.

Input data of unsupervised learners are unlabeled; it is hard to evaluate the accuracy of the structure that is output by the relevant algorithms, which can be distinguished from supervised learning.

### Related Work

Another structure of unsupervised abnormality NIDS dependent on the exception recognition strategy in irregular woods calculation [17]. The structure manufactures the examples of system benefits over datasets named by the administrations. With the fabricated examples, the system distinguishes assaults in the datasets utilizing the altered exception location calculation, lessening the count unpredictability. This methodology is autonomous of assault free preparing datasets, yet accepts that each system administration has its very own example for typical exercises.

A novel component portrayal approach, to be specific the bunch focus and closest neighbor (CANN) approach [18]. In this methodology, two separations are estimated and summed, the first dependent on the separation between every datum test and its bunch focus, and the second separation is between the information and its closest neighbor in a similar group. At that point, this new and one-dimensional

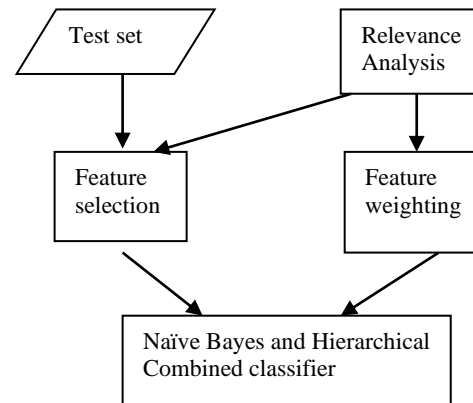
separation-based element is utilized to speak to every datum test for interruption discovery by a k-Nearest Neighbor (k-NN) classifier. This methodology can unquestionably improve the run time with the one-dimensional component.

Decision trees (DT) and support vector machines (SVM) are consolidated as a various leveled crossover shrewd framework display (DTSVM) and a gathering approach joining the base classifiers [19]. In this methodology, they first build DT, SVM and half and half DTSVM classifiers independently to acquire a decent speculation exhibition. Test information is gone through every individual model and the comparing yields are utilized to choose the last yield. The proposed recognition display joins the individual base classifiers and other mixture AI standards to augment identification precision and limit computational multifaceted nature.

An alternate hereditary calculation which can be utilized to prepare the feed forward neural system which is utilized to recognize the interruption adequately [20]. The GA-weight calculation is utilized in this way to deal with persuade upgraded loads to be utilized by the NN, at that point the NN arranges the given standardized system traffic into typical and strange classes and furthermore give the precision of the expectation.

## PROPOSED METHOD

In this paper, we are proposing to improve Naïve Bayes grouping by presenting highlight choosing and weighting, and using unsupervised learning strategy various leveled bunching. The proposed strategy principally comprises of two phases (Fig. 1).



**Figure 1: Proposed model.**

We first lead highlight choosing to pick the most significant highlights for assault order and get an ideal dataset for approval. At that point distinctive loads are relegated to the chose highlights dependent on the significance among highlights and assault classes. The most applicable highlights are relegated higher loads, so that to make the classifier progressively delicate to ordinary assaults. Next, we proposed a Naïve bayes and various leveled bunching joined calculation to improve the execution of classifier. The highlights loads are connected while experiencing the proposed calculation.

Highlight choosing and weighting technique: According to the significance of highlights, we can reason that there are numerous highlights that can separate assaults effectively, however there are likewise repetitive highlights with little data addition and a few highlights even demonstrate no varieties in the preparation set. The little data addition of these highlights demonstrates that they contribute little for interruption discovery.

Therefore, we conduct a feature selecting method based on analyzing the involvement of each feature to classification and we only select the most relevant features to get an optimal 19-dimension feature set, so as to reduce the feature dimensions and discard useless features. The selected features and their corresponding descriptions are shown in Table 1.

**Table 1: Network attributes.**

| Sr. No. | Network Attributes          |
|---------|-----------------------------|
| 1       | Duration                    |
| 2       | Protocol_Type               |
| 3       | SERVICE                     |
| 4       | FLAG                        |
| 5       | SRC bytes                   |
| 6       | DST bytes                   |
| 7       | Land                        |
| 8       | Wrong Fragment              |
| 9       | Urgent                      |
| 10      | Hot                         |
| 11      | num failed logins           |
| 12      | logged_in                   |
| 13      | numcompromised              |
| 14      | Rootshell                   |
| 15      | su_attempted                |
| 16      | Numroot                     |
| 17      | Numfilecreations            |
| 18      | num_shells                  |
| 19      | numaccessfiles              |
| 20      | Numoutboundcmds             |
| 21      | is_host_login               |
| 22      | is_guest_login              |
| 23      | Count                       |
| 24      | srv_count                   |
| 25      | error_rate                  |
| 26      | srv_error_rate              |
| 27      | rerror_rate                 |
| 28      | srv_rerror_rate             |
| 29      | same_srv_rate               |
| 30      | diff_srv_rate               |
| 31      | srv_diff_host_rate          |
| 32      | dst_host_count              |
| 33      | dst_host_srv_count          |
| 34      | dst_host_same_srv_rate      |
| 35      | dst_host_diff_srv_rate      |
| 36      | dst_host_same_src_port_rate |
| 37      | dst_host_srv_diff_host_rate |
| 38      | dst_host_rerror_rate        |
| 39      | dst_host_srv_rerror_rate    |
| 40      | dst_host_rerror_rate        |
| 41      | dst_host_srv_rerror_rate    |

Moreover, DoS and Probe attacks usually reveal a pattern that is different from normal examples so they can be easily differentiated from normal examples. However, U2R and R2L attacks do not reveal a similar pattern and they are embedded in the contents of the packets. So, the U2R and R2L attacks are very difficult to classify in terms of detection,

which is more challenging than Dos and Probe attacks. So, we introduce feature weights in our work in order to properly classify these attacks. By assigning weights to relevant features, we can make the distance between examples more sensitive to relevant features. The most relevant features we found for each class are shown in Table 2.



## Naïve Bayesian and Hybrid clustering Combined Algorithm

**Table 2:** Data used for training and testing in the experiments.

| Category         | Training Set   | Test Set   |
|------------------|--|--|
| DOS<br>(391,458) | Normal 40,000, Smurf 10,000, Neptune 5000, back 1000, land 10, pod 100, teardrop 400                           | Normal 40,000, Smurf 10,000, Neptune 5000, back 1203, land 11, pod 164, teardrop 579             |
| Probe<br>(4107)  | Normal 40,000, satan 800, portsweep 500, nmap 110, insweep 600   | Normal 40,000, satan 789, portsweep 540, nmap 121, insweep 647                                   |
| R2L<br>(1126)    | Normal 40,000, FTP write 4, guess passwd 23, imap 7, multihop 3, warezclient 520, warezmaster 10, phf 4. Spy 2 | Normal 40,000, FTP write 4, guess passwd 30, imap 5, multihop 4, warezclient 500, warezmaster 10 |
| U2R<br>(52)      |  | Normal 40,000, buffer_overflow 15, motkit 6, load module 5, perl 3                               |

Our proposed method based on the Hybrid clustering algorithm, and utilize the unsupervised learning, Naive Bayesian to improve the attack classification performance.

### Motivation and Theory

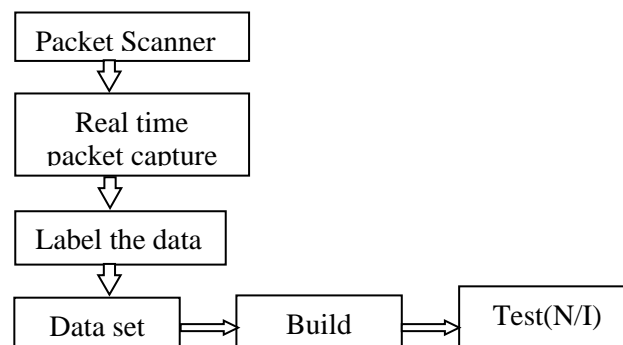
Naive bayes Rule is the basis for many machine-learning and data mining methods. This algorithm is used to create models with predictive capabilities. It provides new ways of exploring and understanding data. Generally, Naive bayes classifier technique is used when the data is high and when the attributes are independent of each other [18]. Naive bayes classifier algorithm is used to model normal and suspicious network activity. The Naive bayes classifier is a supervised learning algorithm based largely off of bayes theorem:

$$P(B/A) = P(A/B) P(B)P(A)$$

We can calculate the probability that an attack is occurring based on some data by first calculating the probability that some

previous data was part of that type of attack and then multiplying by the probability of that type of attack occurring In this paper we are considering 3 attacks and 1 normal behavior of the packet as follows:

1. A SYN flood is a form of denial-of-service attack in which an attacker sends a succession of SYN requests to a target's system in an attempt to consume enough server resources to make the system unresponsive to legitimate traffic.
2. A UDP flood attack is a denial-of-service attack that can be initiated by sending a large number of UDP packets to random ports on a remote host.
3. A TCP Data flood is the denial of service attack in which an attacker sends TCP data as fast as the air interface will allow.
4. Normal connections are generated by simulated daily user behavior. To solve this problem, we introduce Hierarchical clustering (Fig. 2).



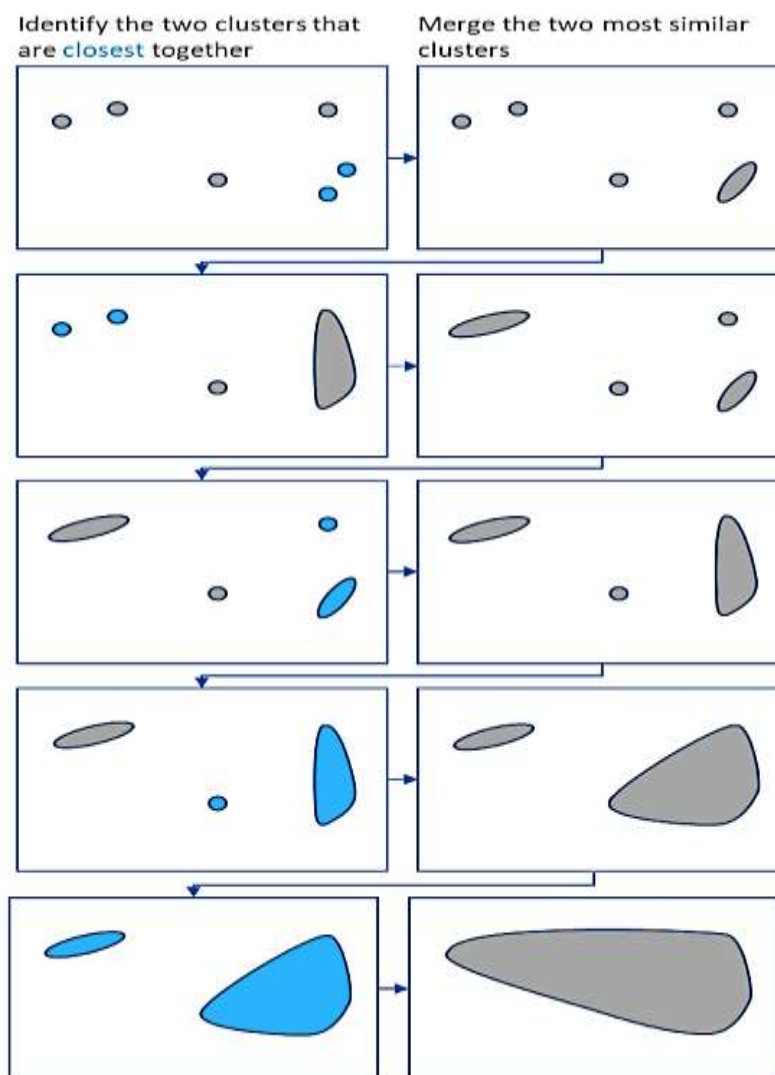
**Figure 2:** Hierarchical clustering.

### MOST RELEVANT FEATURES FOR ATTACK TYPES

| Attacks | Methods        | Attributes selected                           |
|---------|----------------|---|
| DOS     | IG             | 5, 23, 3, 24, 6, 2, 36 (ranking)              |
|         | Wrapper (BN)   | 4, 5, 8, 10, 13, 23, 37                       |
|         | Wrapper (C4.5) | 3, 5, 6, 13, 23                               |
| Probe   | IG             | 5, 3, 6, 35, 33, 34, 4, 27, 23 (ranking)      |
|         | Wrapper (BN)   | 3, 4, 5, 29, 32, 35                           |
|         | Wrapper (C4.5) | 5, 29, 30, 35, 39, 40                         |
| R2L     | IG             | 5, 3, 6, 33, 36, 10, 37, 24, 1                |
|         | Wrapper (BN)   | 1, 5, 6, 22, 23, 32                           |
|         | Wrapper (C4.5) | 1, 3, 5, 6, 12, 31                            |
| U2R     | IG             | 3, 33, 13, 14, 1, 10, 5, 17, 32, 36 (ranking) |
|         | Wrapper (BN)   | 1, 2, 5, 14, 36                               |
|         | Wrapper (C4.5) | 1, 13, 14, 32                                 |

**Hierarchical Clustering:** Also known as hierarchical cluster analysis, is an algorithm that groups similar objects into groups called *clusters*. The endpoint is a

set off clusters, where each cluster is distinct from each other cluster, and the objects within each cluster are broadly similar to each other (Fig. 3).



**Figure 3:** Clustering.

The calculation we proposed is depicted as pursues. Right off the bat, we direct bunching utilizing Naïve Bayesian grouping calculation on the pre-handled testing set. The quantity of groups is chosen by the quantity of assault classes in the dataset. The KDD datasets, for instance, contains Dos, Probe, U2R, R2L and Normal absolutely five sorts of information, so the quantity of bunches is set to 5. Likeness of each class is iteratively registered. At that point for a test precedent in the testing set, the neighbors are reselected dependent on comparability.

## CONCLUSION

This paper right off the bat include choice and weighting approach dependent on estimating the importance among highlights and assault classes. By thusly we dole out various loads to the most important highlights and make our proposed calculation increasingly compelling for assault grouping. We proposed a proficient interruption discovery strategy that joins administered learning Naïve Bayesian arrangement and unsupervised learning Hierarchical grouping for assault order task, in which the neighbors are reselected and arranged dependent on the closeness to their relating bunches. The new neighbors vote the last grouping outcomes.

## REFERENCES

1. KDD data set (1999), Available from <http://kdd.ics.uci.edu/databases/>.
2. Sabhnani M, Serpen G (2004), "Why Machine Learning Algorithms Fail in Misuse Detection on KDD Intrusion Detection Data Set", *In Journal of Intelligent Data Analysis*, Volume 8, Issue 4, pp. 403-415.
3. HG Kayacik, AN Zincir-Heywood, MJ Heywood (October 2005), "Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets", *Proc. 3rd Annual Conference on Privacy Security and Trust*.
4. AK Jain, RPW Duin, J Mao (2000), "Statistical pattern recognition: a review", *IEEE Trans. Pattern Anal. Mach. Intell.*, Volume 22, Issue 1, pp. 4-37.
5. H Altwaijry, S Algarny (2012), "Bayesian based intrusion detection system", *Journal of King Saud University - Computer and Information Sciences*, Volume 24, Issue 1, pp.1-6.
6. P Somwang, W Lilakiatsakun (2011), "Computer network security based on Support Vector Machine approach", *11th International Conference on Control, Automation and Systems, (ICCAS 2011)*.
7. CA Catania, F Bromberg, CG Garino (2012), "An autonomous labeling approach to support vector machines algorithms for network traffic anomaly detection", *Expert Systems with Applications*, Volume 39, Issue 2, pp. 1822-1829.
8. N Kausar, BB Samir, SB Sulaiman, I Ahmad, M Hussain (2012), "An approach towards intrusion detection using PCA feature subsets and SVM", *2012 International Conference on Computer & Information Science (ICCIS)*.
9. W Li, Z Liu (2011), "A method of SVM with Normalization in Intrusion Detection", *Procedia Environmental Sciences*, Volume 11, Part A (0), pp. 256-262.
10. SM Lee, DS Kim, JH Lee, JS Park (2012), "Detection of DDoS attacks using optimized traffic matrix", *Computers & Mathematics with Applications*, Volume 63, Issue 2, pp. 501-510.
11. PK Sujatha, CS Priya, A Kannan (2012), "Network intrusion detection system using genetic network programming with support vector machine", *Proceedings of the International Conference on Advances*



- in Computing, Communications and Informatics*. pp. 645–649, Chennai, India.
12. Y Li, L Guo (2007), “An active learning based TCMKNN algorithm for supervised network intrusion detection”, *Computer and Security*, Volume 26, Issues 7-8, pp. 459–467.
  13. HM Shirazi (2009), “Anomaly Intrusion Detection System Using Information Theory, K-NN and KMC algorithms”, *Australian Journal of Basic & Applied Sciences*, Volume 3, Issue 3, pp. 2581–2597.
  14. C Zhang, J Jiang, M Kamel (2005), “Intrusion detection using hierarchical neural network”, *Pattern Recogn. Lett.*, Volume 26, Issue 6, pp. 779–791.
  15. S Ganapathy, K Kulothungan, P Vogesh, A Kannan (2012), “A Novel Weighted Fuzzy C Means Clustering Based on Immune Genetic Algorithm for Intrusion Detection”, *Procedia Engineering*, Volume 38, pp. 1750–1757.
  16. Jiong Zhang, Mohammad Zulkernine (2006), “Anomaly Based Network Intrusion Detection with Unsupervised Outlier Detection”, *IEEE International Conference on Communications*.
  17. Wei-Chao, Lin. (2015), “CANN: An intrusion detection system based on combining cluster centers and nearest neighbors”, *Knowledge-Based Systems*, Volume 78, pp. 13–21.
  18. Sandhya Peddabachigari (2007), “Modeling intrusion detection system using hybrid intelligent systems”, *Journal of Network and Computer Applications*, Volume 30, Issue 1, pp. 114–132.
  19. P Srinivasu (2012), “Genetic Algorithm based Weight Extraction Algorithm for Artificial Neural Network Classifier in Intrusion Detection”, *Procedia Engineering*, Volume 38, pp. 144–153.
  20. Z Zhuo, Y Zhang, Z Zhang, X Zhang, J Zhang (May 2018), “Web-site Fingerprinting Attack on Anonymity Networks Based on Profile Hidden Markov Model”, *IEEE Transactions on Information Forensics and Security*, Volume 13, Issue 5, pp. 1081–1095.

***Cite this article as:***