Contents lists available at Science-Gate

International Journal of Advanced and Applied Sciences

Journal homepage: http://www.science-gate.com/IJAAS.html

Detecting block ciphers generic attacks: An instance-based machine learning method

Yazan Ahmad Alsariera *

Department of Computer Science, College of Science, Northern Border University, Arar, Saudi Arabia

ARTICLE INFO

Article history:
Received 15 December 2021
Received in revised form
3 February 2022
Accepted 3 March 2022

Keywords:
Cryptography
Generic attack
Instance-based machine learning
Cybersecurity and intrusion detection

ABSTRACT

Cryptography facilitates selective communication through encryption of messages and or data. Block-cipher processing is one of the prominent methods for modern cryptographic symmetric encryption schemes. The rise in attacks on block-ciphers led to the development of more difficult encryption schemes. However, attackers decrypt block-ciphers through generic attacks given sufficient time and computing. Recent research had applied machine learning classification algorithms to develop intrusion detection systems to detect multiple types of attacks. These intrusion detection systems are limited by misclassifying generic attacks and suffer reduced effectiveness when evaluated for detecting generic attacks only. Hence, this study introduced and proposed *k*-nearest neighbors, an instance-based machine learning classification algorithm, for the detection of generic attacks on block-ciphers. The value of *k* was varied (i.e., 1, 3, 5, 7, and 9) and multiple nearest neighbors classification models were developed and evaluated using two distance functions (i.e., Manhattan and Euclidean) for classifying between generic attacks and normal network packets. All nearest neighbors models using the Manhattan distance function performed better than their Euclidean counterparts. The 1-nearest neighbor (Manhattan distance function) model had the highest overall accuracy of 99.6%, a generic attack detection rate of 99.5% which tallies with the 5, 7, and 9 nearest neighbors models, and a false alarm rate of 0.0003 which is the same for all Manhattan nearest neighbors classification models. These instance-based methods performed better than some existing methods that even implemented an ensemble of deep-learning algorithms. Therefore, an instance-based method is recommended for detecting block-ciphers generic attacks.

1. Introduction

Cryptography is the art of coding messages or information to facilitate selective communication (Bhattacharyya and Chakrabarti, 2022). In other words, it is the art and science of introducing secrecy into information security (Samoriski, 2020). Traditionally, cryptography involves manipulating letters or digits and it is based on providing security through obscurity (Aswath et al., 2022). Traditional block cipher involves the encryption of data via the manipulation of letters and digits (Nahar and Chakraborty, 2020). Traditional ciphers are usually encrypted using the symmetric key encryption method, which uses the same key for encryption and decryption (Kshirsagar and Shah, 2021). However, modern cryptography is built on the concepts of mathematics number theory, probability theory, and computational complexity theory for the encryption of data (Saračević et al., 2020). It deals with the security of digital data which is represented as strings of binary digits (Easttom, 2021). Modern cryptography utilizes one of those mathematical concepts to convert strings of plain binary digits into coded binary strings for encryption to take place. Modern symmetric encryption schemes are categorized into block ciphers and stream ciphers based on how plain binary strings are processed (Sevin and Mohammed, 2021). This study is interested in block ciphers.

Block ciphers encryption scheme processes plain strings of binary text in blocks (i.e., groups) of bits at a time. For example, the modern Advanced Encryption Standard (AES) scheme processes 128

* Corresponding Author.
Email Address: yazan.sadeq@nbu.edu.sa
https://doi.org/10.21833/ijaas.2022.05.007
Corresponding author's ORCID profile:
https://orcid.org/0000-0003-1359-6336

bits of a plain string of binary texts at a time (Awan et al., 2020). Block ciphers are developed by selecting a block of plain strings of bits, performing an encryption function on the selected bits, generating a block of block cipher, and repeating these processes until all plain strings of bits are transformed into their corresponding bits of block ciphers (Bahadori et al., 2021).

To protect data integrity, cryptographic experts developed various complicated schemes of block ciphers to deter attackers from decrypting block ciphers. However, with enough computational resources, most block ciphers can be decrypted (Shetty et al., 2020). Attackers are usually aware that data or information is being communicated or transmitted, although they are encrypted or scrambled messages. Thus, they usually intrude on a digital network to attack encrypted data or information. One of these dangerous attacks is referred to as a 'generic attack on block ciphers'. A generic attack on block ciphers (Moustafa and Slay, 2015; Kumar et al., 2020) usually involves running a brute-force attack on the block ciphers regardless of the encryption structure of such block ciphers.

Cybersecurity aims to ensure the integrity, confidentiality, and availability of data, information, and resources (Gauthama Raman et al., 2020) across cyberspace consisting of billions of connected users and devices (Faker and Dogdu, 2019). Hence, various countermeasures against generic attacks are being developed (Dutta et al., 2019). One of the prominent countermeasures is the use of classification machine learning algorithms for detecting intrusions (attacks) (Wei et al., 2020). Intrusion detection systems detect network packets as a normal network packets or one of the various forms of attacks (Idhammad et al., 2018; Alsariera et al., 2020a; 2020b; 2021a; 2021b). Most of the existing multi-classification intrusion detection systems suffer from the misclassification of attacks that shares the same characteristics (Salman et al., 2017). Salman et al. (2017) demonstrated how multiclassification intrusion detection system tends to misclassify generic attack as exploited attacks at about 51% error. Therefore, it becomes essential to develop intrusion detection models specifically for the detection of the nefarious generic attacks on block ciphers to appropriately defend against this form of attack.

To develop such an intrusion detection model, data becomes pivotal. Most of the publicly available network data such as NSL-KDD intrusion network data (i.e., an improved KDDCup'99) does not capture generic attacks (Xin et al., 2018). The UNSW-NB15 dataset (Moustafa and Slay, 2015) captured generic attacks and other forms of attacks. Meanwhile, the UNSW-NB15 dataset is usually used in research for developing multi-classification (Nawir et al., 2018) or anomaly (Feng et al., 2019) intrusion detection models. Hence, this study aims to develop a generic attack detector through the implementation of an instance-based machine learning classification algorithm. As such, the contributions to knowledge made by this research include introducing an instance-based machine learning classification algorithm to detect generic attacks at a higher detection rate and lower false alarm rate. Another contribution of this study is conducting a robust empirical analysis of various instance-based classification models to identify the best-performing model(s) to classify between generic attacks and normal packets.

The remaining sections of this study include the review of related works, methodology, results, discussion, conclusion, and future works.

2. Review of related works

There is more research on multi-classification models for detecting generic attacks than stand-alone generic attack detectors. However, most of the research on multi-classification methods does not report the performance of their model for each type of attack. In this study, we reviewed some of the published works on multi-classification models for detecting attacks that provided the performance of their model for generic attacks.

Thaseen et al. (2020a; 2020b) identified the performance results of implementing an integration of a majority voting ensemble of long-short-term memory deep learning method and embedded feature extraction module to detect generic attacks and other forms of attacks in a multi-classification method. The original performance of this method was a 99.9% overall accuracy for multiclassification which reduced to 95.23% accuracy for detecting generic attacks.

Another study by Gharaee and Hosseinvand (2017) developed a genetic algorithm to select the best variables to detect attacks and used a support vector machine learning algorithm to fit a multi-classification model to detect generic and other forms of attacks contained in the UNSW-NB15 datasets. The study provided the specific performance of its method for detecting generic attacks. Their method was reported to detect generic attacks at a 96.69% true positive rate, misclassified normal packets, and other attacks as generic attacks at a 0.01% false alarm rate, and resulted in an overall accuracy of 97.51%.

Olasehinde (2020) implemented k-Nearest neighbor, Naïve Bayes, and Decision Tree classification algorithms as a base-learner for three different implementations of stacked ensemble methods. The stacked ensemble methods were Multiple Model Trees (MMT), Meta Decision Trees (MDT), and Multi-Response Linear Regression (MLR). This study reported the performance of various stacked ensemble models with the integration of the feature selection method for multi-classification rather than each attack. It was reviewed as an instance-based method (i.e., k-nearest neighbors) and was included as a based learner. The MMT ensemble method produced 96.89% overall accuracy, the MLR method had

97.8% overall accuracy, and the MDT ensemble method had 98.08% overall accuracy.

Kumar et al. (2020) published a novel rule-based multi-classification method on the Generic, DOS, Exploit, and Probe attacks contained in the UNSW-NB15 dataset. The performance of the rule-based method was reported to be an overall average accuracy of 65.21{% for all classes of attacks and a False Alarm Rate of 2.01%.

Through the review of literature, multi-classification models are seen to have reduced effectiveness in detecting generic attacks. More so, the performances of the multi-classification models need improvements considering the dangerous effect of the successful execution of generic attacks.

3. Method

3.1. Dataset

This study considers the UNSW-NB15 dataset as it contains contemporary normal network packets and generic attacks among others (Moustafa and Slay, 2015). The KDDCup'99 (Li et al., 2018) and NSL-KDD datasets do not contain the attack considered in this study (Mabayoje et al., 2016; Saleh et al., 2019).

The data used in this study is a balanced extraction of all generic attack instances in the UNSW-NB15 dataset and enough normal packet instances. This developed dataset contains all features of the original dataset besides the features 'id' and 'attack_cat' which are not relevant to this study. Therefore, the developed data contain forty-two (42) independent features and one dependent feature with two values (i.e., generic and normal).

Hence, the developed data for this study contained 18,871 generic attack instances and 18,954 normal packets.

3.2. Implemented models

The instance-based classification machine learning algorithm is also referred to as memory-based reasoning, lazy learning, example-based reasoning, or case-based reasoning (Verma and Shakya, 2021). It is one of the available non-parametric categories of machine learning algorithms. It does not assume the inherent data distribution to develop a classification model rather it waits until the testing phase to compute the class an instance belongs to Mabayoje et al. (2019) and Sharma et al. (2019).

Although the Nearest neighbor algorithm can be used for regression and classification, it is implemented and used as a classification algorithm for detecting generic attacks on a network as befitting to this study. Regarding classification, the Nearest Neighbor algorithm simply classifies an instance based on its distance to the specified number of nearest instance(s) as illustrated in Pseudocode 1.

Algorithm 1: k-Nearest Algorithm

Let (X_i, C_i) where i = 1, 2, …, n be data points.
X_i denotes feature values and C_i denotes labels for Xi for each i.
Assuming the number of classes as 'c', $c_i \in [1,2,3, …, c]$ for all values of i
Let x be a point for an unknown label, and k-NN find the label.

1: Calculate $\boldsymbol{d}(x, x_i)$, i = 1, 2, …,**n**; where ***d*** represent the distance between those points.
2: Arrange the calculate **n** distance in non-decreasing order
3: Let ***k*** be a positive number, select the first ***k*** distances from number 2 above.
4: Find those ***k***-points corresponding to the selected ***k***-distances
5: Assign class *i* to x, where *i* is the majority label of the selected ***k***-distances

Theoretically, Nearest Neighbor assumes that data is in a feature space and its instances (or data points) are at distance among themselves. Each data instance is made up of independent variables and a class label. Also, it assumes a single positive number "k" is given which determines the number of neighbors useful for classification.

Given the fact that the value of *k* is pivotal in implementing a Nearest Neighbor classification algorithm and there are two class labels in the 'Generic' attack dataset, this study considered evaluating the odd values contained within the range of 1 to 10 (i.e., 1, 3, 5, 7 and 9).

A typical 1-Nearest Neighbor classification model assigns the class label of the closest instance to the predicted instance. The other k-Nearest Neighbor classification models assign most of the class labels of k instances to the predicted instance. Therefore, this study implemented 1-Nearest Neighbor and four different types of k-Nearest Neighbor to classify network traffics into either normal traffic or generic attack.

The experimental framework of this study is graphically depicted in Fig. 1. All generic network attack instances were extracted from the UNSW-NB15 training dataset to develop a dataset. An adequate number of normal traffic instances were also extracted and appended to the data to form a balanced dataset. The balanced dataset was randomly shuffled to mingle the generic attack instances and normal traffic instances within the dataset. Two distance functions were implemented (i.e., Euclidean and Manhattan) for each instance-based method before model development. Each nearest neighbors method was fitted on the randomly shuffled dataset via 10-fold cross-validation. The 10-fold cross-validation technique fits a robust model by splitting the dataset into 10 partitions. It trains the model using the first 9 splits and tests on the set-aside split. This is repeated 10 times until all splits are used for training and testing.

The 10 models are then aggregated to produce a robust model. The total number of generic attacks and normal traffic instances that were correctly and falsely classified by the fitted models (i.e., 1, 3, 5, 7, 9-Nearest Neighbor classification models) were reported as a confusion matrix.

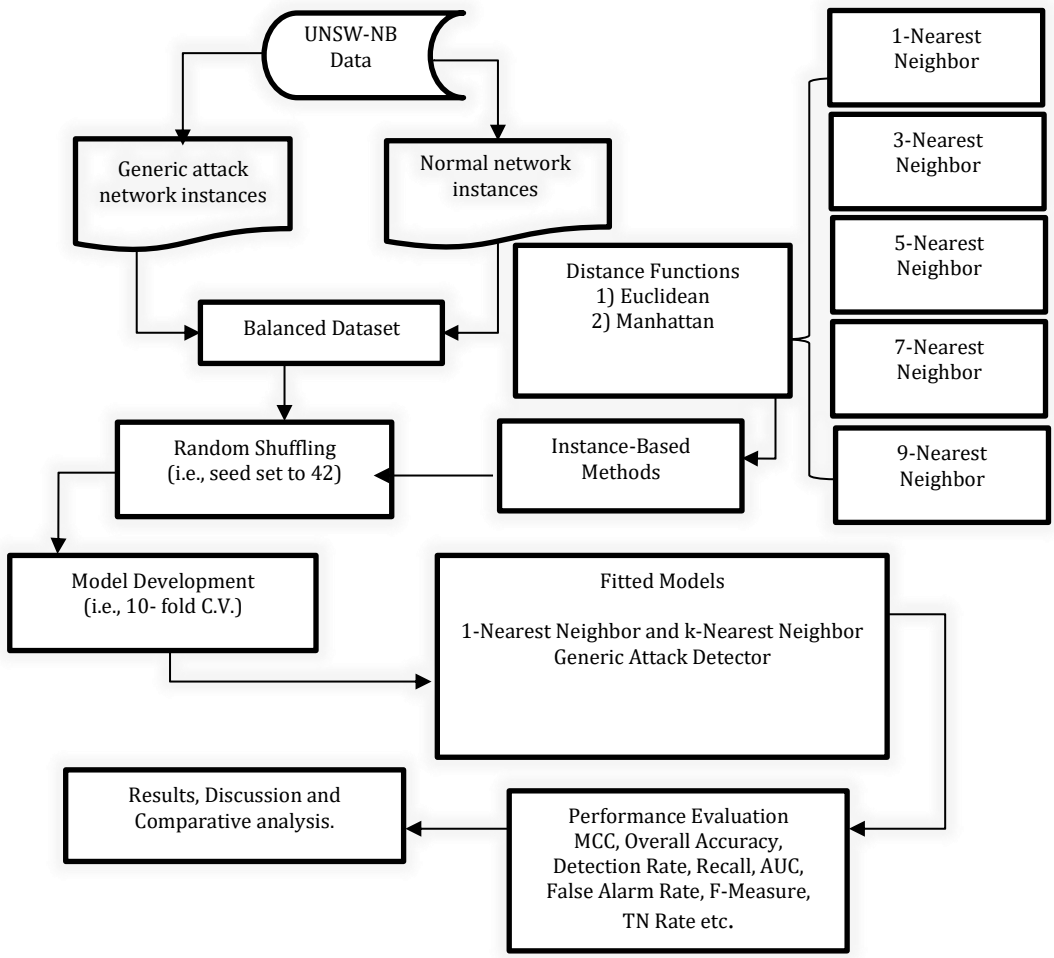

Fig. 1: Experimental framework

3.3. Performance evaluation metrics

This study aims to develop instance-based machine learning models for classifying between generic attacks and normal traffic. This is typically a binary type of classification (i.e., two class values) model. Using the populated values in the confusion matrix, (i.e., True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN)), the performances of the proposed instance-based methods were evaluated by deriving these evaluation scores, namely: Matthews Correlation Coefficient (MCC), True Positive (TP) Rate (i.e., Detection Rate), False Positive (FP) Rate (i.e., False Alarm Rate), F-Measure, and Area Under Curve (AUC) (Elijah et al., 2019; Alsariera et al., 2020a; 2020b; 2021a; 2021b; Kasongo and Sun, 2020; Mebawondu et al., 2020; Sarumi et al., 2020; Thaseen et al., 2020a; 2020b). Additionally, the kappa value and overall accuracy (i.e., the percentage of correctly classified 'Generic' attack and normal network traffic) were calculated for each instance-based model.

Through the review of literature, the MCC score is optimal for evaluating binary classification models because it uses all the populated counts or values in the confusion matrix table (Li et al., 2020; Thaseen et al., 2020a; 2020b). MCC metric is calculated as seen in Eq. 1.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Eq. 1 shows Matthews Correlation Coefficient (Li et al., 2020).

MCC measure tells the correlation coefficient among the detected and expected predictions, and its value ranges from 0 to 1 (Mebawondu et al., 2020). In this study, the MCC value of each instance-based model is used to compare the performance of each model alongside other performance evaluation measures.

4. Results

The 1-Nearest Neighbor model for detecting generic attacks using the Euclidean distance function correctly classified 18,773 of 18,871 generic attacks and 18,869 of 18,954 normal traffic (Table 1).

Table 1: Confusion matrix for 1-nearest neighbor (Euclidean distance)

	Generic	Normal
Generic	18,773	98
Normal	85	18,869

From Table 1, the 1-Nearest Neighbor model with Euclidean distance falsely classified 98 generic attacks as normal traffic, and 85 normal traffic were falsely classified as generic attacks. The 1-Nearest Neighbor model for detecting generic attacks using the Manhattan distance function correctly classified 18,783 of 18,871 generic attacks and 18,895 of 18,954 normal traffic. Table 2 shows the confusion matrix for 1-Nearest Neighbor (Manhattan Distance).

Table 2: Confusion matrix for 1-nearest neighbor (Manhattan distance)

	Generic	Normal
Generic	18,783	88
Normal	59	18,895

From Table 2, the 1-Nearest Neighbor model with Manhattan distance falsely classified 88 generic attacks as normal traffic and just 59 normal traffic instances as generic attacks. The values of the derived performance measures are revealed in Table 3.

Table 3: Performance scores of 1-NN models

Evaluation Metric	1-NN Euclidean Distance	1-NN Manhattan Distance
Accuracy	99.52%	99.61%
Kappa	0.9903	0.9922
Detection Rate	0.995	0.995
False Alarm Rate	0.004	0.003
F-measure	0.995	0.996
MCC	0.99	0.992
AUC	0.997	0.998

"NN": Nearest Neighbor

The comparison of performance scores of the 1-Nearest Neighbor models reveals that the 1-Nearest Neighbor model developed via Manhattan distance is better for detecting generic attacks. With higher overall accuracy, lower false alarm rate, and a higher MCC value among others, the Manhattan distance 1-Nearest Neighbor classification model for detecting generic attacks in the presence of normal network traffic is better than the Euclidean distance 1-Nearest Neighbor.

The 3-Nearest Neighbor model for detecting generic attacks using the Euclidean distance function correctly classified 18,763 of 18,871 generic attacks and 18,881 of 18,954 normal traffic (Table 4).

Table 4: Confusion matrix for 3-nearest neighbor (Euclidean distance)

	Generic	Normal
Generic	18,765	106
Normal	73	18,881

From Table 4, the 3-Nearest Neighbor model with Euclidean distance falsely classified 106 generic attacks as normal traffic while 73 normal traffic were falsely classified as generic attacks. The 3-Nearest Neighbor model for detecting generic attacks using the Manhattan distance function correctly classified 18,765 of 18,871 generic attacks and 18,906 of 18,954 normal traffic. Table 5 shows confusion matrix for 3-nearest neighbor (Manhattan distance).

Table 5: Confusion matrix for 3-nearest neighbor (Manhattan distance)

	Generic	Normal
Generic	18,765	106
Normal	48	18,906

From Table 5, the 3-Nearest Neighbor model with Manhattan distance falsely classified 106 generic attacks as normal traffic and just 48 normal traffic instances as generic attacks. The values of the performance measures derived from the confusion matrix are revealed in Table 6.

Table 6: Performance scores of 3-NN models

Evaluation Metric	3-NN Euclidean Distance	3-NN Manhattan Distance
Accuracy	99.53%	99.59%
Kappa	0.9905	0.9919
Detection Rate	0.994	0.994
False Alarm Rate	0.004	0.003
F-measure	0.995	0.996
MCC	0.991	0.992
AUC	0.999	0.999

"NN": Nearest Neighbor

The comparison of performance scores of the 3-Nearest Neighbor models is like the results of 1-Nearest Neighbor classification models where the Manhattan distance Nearest Neighbor had the better performance.

The 5-Nearest Neighbor model for detecting generic attacks using the Euclidean distance function correctly classified 18,762 of 18,871 generic attacks and 18,877 of 18,954 normal traffic (Table 7).

Table 7: Confusion matrix for 5-nearest neighbor (Euclidean distance)

	Generic	Normal
Generic	18,762	109
Normal	77	18,877

From Table 7, the 5-Nearest Neighbor model with Euclidean distance falsely classified 109 generic attacks as normal traffic, and 77 normal traffic were falsely classified as generic attacks. The 5-Nearest Neighbor model for detecting generic attacks using the Manhattan distance function correctly classified 18,769 of 18,871 generic attacks and 18,899 of 18,954 normal traffic. Table 8 shows confusion matrix for 5-nearest neighbor (Manhattan distance).

Table 8: Confusion matrix for 5-nearest neighbor (Manhattan distance)

	Generic	Normal
Generic	18,769	102
Normal	55	18,899

From Table 8, the 5-Nearest Neighbor model with Manhattan distance falsely classified 102 generic attacks as normal traffic and 55 normal traffic instances as generic attacks. The values of the performance measures derived from the confusion matrix are revealed in Table 9.

The comparison of performance scores of the 5-Nearest Neighbor models is like the 3-Nearest Neighbor classification models where the Manhattan

distance Nearest Neighbor had the better performance.

The 7-Nearest Neighbor model for detecting generic attacks using the Euclidean distance function correctly classified 18,761 of 18,871 generic attacks and 18,878 of 18,954 normal traffic (Table 10).

Table 9: Performance scores of 5-NN models

Evaluation Metric	5-NN Euclidean Distance	5-NN Manhattan Distance
Accuracy	99.51%	99.59%
Kappa	0.9905	0.9917
Detection Rate	0.994	0.995
False Alarm Rate	0.004	0.003
F-measure	0.995	0.996
MCC	0.99	0.992
AUC	0.999	0.999

"NN": Nearest Neighbor

Table 10: Confusion matrix for 7-nearest neighbor (Euclidean distance)

	Generic	Normal
Generic	18,761	110
Normal	76	18,878

From Table 10, the 7-Nearest Neighbor model with Euclidean distance falsely classified 110 generic attacks as normal traffic, and 76 normal traffic were falsely classified as generic attacks. The 7-Nearest Neighbor model for detecting generic attacks using the Manhattan distance function correctly classified 18,770 of 18,871 generic attacks and 18,898 of 18,954 normal traffic. Table 11 shows confusion matrix for 7-nearest neighbor (Manhattan distance).

Table 11: Confusion matrix for 7-nearest neighbor (Manhattan distance)

	Generic	Normal
Generic	18,770	101
Normal	56	18,898

From Table 11, the 7-Nearest Neighbor model with Manhattan distance falsely classified 101 generic attacks as normal traffic and 56 normal traffic instances as generic attacks. The values of the performance measures derived from the confusion matrix are contained in Table 12.

Table 12: Performance scores of 7-NN models

Evaluation Metric	7-NN Euclidean Distance	7-NN Manhattan Distance
Accuracy	99.51%	99.59%
Kappa	0.9905	0.9917
Detection Rate	0.994	0.995
False Alarm Rate	0.004	0.003
F-measure	0.995	0.996
MCC	0.99	0.992
AUC	0.999	0.999

"NN": Nearest Neighbor

The 7-Nearest Neighbor (Manhattan Distance) classification model performed better than its Euclidean distance counterpart.

The 9-Nearest Neighbor model for detecting generic attacks using the Euclidean distance function correctly classified 18,760 of 18,870 generic attacks and 18,873 of 18,873 normal traffic (Table 13).

Table 13: Confusion matrix for 9-Nearest Neighbor (Euclidean Distance)

	Generic	Normal
Generic	18,760	111
Normal	81	18,873

The 9-Nearest Neighbor model with Euclidean distance falsely classified 111 generic attacks as normal traffic and 81 normal traffic were falsely classified as generic attacks. On the other hand, the 9-Nearest Neighbor model for detecting generic attacks using the Manhattan distance function correctly classified 18,774 of 18,871 generic attacks and 18,898 of 18,954 normal traffic. Table 14 shows the confusion matrix for 9-Nearest Neighbor (Manhattan Distance).

Table 14: Confusion matrix for 9-Nearest Neighbor (Manhattan Distance)

	Generic	Normal
Generic	18,774	97
Normal	56	18,898

From Table 14, the 9-Nearest Neighbor model with Manhattan distance falsely classified 97 generic attacks as normal traffic and 56 normal traffic instances as generic attacks. The values of the performance measures derived from the confusion matrix are contained in Table 15.

Table 15: Performance scores of 9-NN models

Evaluation Metric	7-NN Euclidean Distance	7-NN Manhattan Distance
Accuracy	99.49%	99.56%
Kappa	0.9898	0.9919
Detection Rate	0.994	0.995
False Alarm Rate	0.004	0.003
F-measure	0.995	0.996
MCC	0.99	0.992
AUC	0.999	1

"NN": Nearest Neighbor

The comparison of performance scores of the 9-Nearest Neighbor models reveals that the 9-Nearest Neighbor model developed via Manhattan distance is better for detecting generic attacks.

5. Discussion

This study aims to develop an optimal instance-based machine learning model capable of detecting generic attacks on block ciphers. The implementation of the proposed experimental framework led to the development of ten (10) instance-based classification models. Five (5) numbers of nearest neighbors values were set (i.e., 1, 3, 5, 7, and 9) and two distance functions were implemented (i.e., Euclidean and Manhattan distance functions). The models were developed using a 10-fold cross-validation model and their classification performances were reported via a confusion matrix. Other performance metrics values were calculated from the confusion matrix values.

The better instance-based classification model (based on the accuracy, detection rate, and false

alarm rate) for each *k* value were all selected and tabulated (Table 16).

Table 16: Comparative analysis of proposed methods

Classification Models	Accuracy (%)	Detection Rate (%)	False Alarm Rate
1-NN (Man.)	99.6114	99.5	0.003
3-NN (Man.)	99.5929	99.4	0.003
5-NN (Man.)	99.5849	99.5	0.003
7-NN (Man.)	99.5849	99.5	0.003
9-NN (Man.)	99.5955	99.5	0.003

"NN": Nearest Neighbor

From the comparative results, all nearest neighbors classification models using the Manhattan distance function are better than their Euclidean distance counterparts across all *k* values for detecting generic attacks and correctly identifying normal network packets. This performance can be safely attributed to how both distance function calculates the distance between two data points.

All nearest neighbors classification models using the Manhattan distance function shared the same MCC value of 0.922. Similarly, they shared the same AUC value of 0.99 except for the 9-nearest Neighbor (Manhattan distance) classification model with an AUC value of 1.0.

The nearest neighbors (Manhattan distance) classification models were all able to classify between normal packets and generic attacks at the lowest accuracy of 99.5849% and the highest accuracy of 99.6114%. These models detected generic attacks at 99.5% except for the 3-nearest neighbors (Manhattan distance) classification model with a detection rate of 99.4%.

Considering the false alarm rate, the nearest neighbors classification models using the Manhattan distance function shared the same value (i.e., 0.003). However, the 3-nearest Neighbor (Manhattan distance) had the lowest number of false positives. This model misclassified only 48 normal traffic as generic attacks (Table 6), at the expense of increasing false negatives (i.e., 106 misclassified generic attacks as normal packets).

Across all Manhattan distance nearest neighbors for classifying between normal network packets and generic attacks, the 1-nearest neighbor's classification model had scored the highest accuracy and higher number of detected generic attacks. However, the 3-nearest neighbor's classification model detected more normal packets and lower false alarm rates. This can be safely translated into a real-time application that for every host on a network under a generic attack, the next host is more likely to be under the same attack. Also, for every 3 closely distanced hosts transmitting normal packets, the fourth host is more likely to transmit packets.

In comparison to existing methods, the instance-based classification method of this study performed better than the reviewed methods. This study's instance-based models detected generic attacks better than the 95.25% detection rate of the majority vote ensemble deep learning method published by Thaseen et al. (2020a; 2020b). This study's instance-based methods detected generic attacks on block-ciphers better than all three stacked ensemble methods of study Olasehinde (2020) with 96.89%, 97.8%, and 98.08% accuracies. The novel rule-based method (Kumar et al., 2020) of 65.21% overall accuracy and 2.01% false alarm rate was outperformed by all implemented instance-based classification models for detecting generic attacks. Given the fact that the instance-based method of this study is specifically trained to detect between normal packets and generic attacks, the successful performance of this study's method is plausible.

6. Conclusion and future works

In conclusion, the aim of this study to introduce, implement and evaluate an instance-based for detecting generic attacks on block ciphers was fulfilled.

The development and evaluation of various nearest neighbors classification methods showed great performance in detecting generic attacks on block ciphers. The overall accuracies of the various methods that were implemented in this study were over 99% while detecting generic attacks at a 99.4% rate at the very least. All nearest neighbors models for detecting generic attacks on block ciphers maintain a low false alarm rate of 0.0003.

In comparison with existing methods, the proposed instance-based methods of this study performed better than all existing multi-classification methods as the study's method is customized to detect generic attacks.

This study does not consider feature selection of the generic attacks variable to investigate if a lesser number of variables can also lead to such high detection performance of generic attacks. Conducting such empirical research to ascertain if feature selection will make or mar the performance of instance-based generic attack detection is the most prominent future work of this study.

Compliance with ethical standards

Conflict of interest

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

Alsariera YA (2021a). Detecting generic network intrusion attacks using tree-based machine learning methods. International Journal of Advanced Computer Science and Applications, 12(2): 597-603. https://doi.org/10.14569/IJACSA.2021.0120275

Alsariera YA (2021b). Hybridized decision tree methods for detecting generic attack on cipher text. International Journal of Computer Science and Network Security, 21(7): 56-62.

Alsariera YA, Adeyemo VE, Balogun AO, and Alazzawi AK (2020a). AI meta-learners and extra-trees algorithm for the detection of phishing websites. IEEE Access, 8: 142532-142542. https://doi.org/10.1109/ACCESS.2020.3013699

Alsariera YA, Elijah AV, and Balogun AO (2020b). Phishing website detection: Forest by penalizing attributes algorithm and its enhanced variations. Arabian Journal for Science and Engineering, 45(12): 10459-10470. https://doi.org/10.1007/s13369-020-04802-1

Aswath S, Valarmathi RS, Mohan Sai Kumar CH, and Pandiyarajan M (2022). Highly secured steganography method for image communication using random byte hiding and confused and diffused encryption. In: Smys S, Bestak R, Palanisamy R, and Kotuliak I (Eds.), Computer networks and inventive communication technologies: 867-884. Springer, Singapore, Singapore. https://doi.org/10.1007/978-981-16-3728-5_65

Awan IA, Shiraz M, Hashmi MU, Shaheen Q, Akhtar R, and Ditta A (2020). Secure framework enhancing AES algorithm in cloud computing. Security and Communication Networks, 2020: 8863345. https://doi.org/10.1155/2020/8863345

Bahadori M, Järvinen K, and Niemi V (2021). FPGA implementations of 256-Bit SNOW stream ciphers for postquantum mobile security. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 29(11): 1943-1954. https://doi.org/10.1109/TVLSI.2021.3108430

Bhattacharyya S and Chakrabarti A (2022). Post-quantum cryptography. In: Sharma N, Chakrabarti A, Balas VE, Bruckstein AM (Eds.), Data management, analytics and innovation: 375-405. Springer, Singapore, Singapore. https://doi.org/10.1007/978-981-16-2937-2_24

Dutta V, Choraś M, Kozik R, and Pawlicki M (2019). Hybrid model for improving the classification effectiveness of network intrusion detection. In the Computational Intelligence in Security for Information Systems Conference, Springer, Seville, Spain: 405-414. https://doi.org/10.1007/978-3-030-57805-3_38

Easttom W (2021). Basic information theory. In: Easttom W (Ed.), Modern cryptography: 51-72. Springer, Cham, Switzerland. https://doi.org/10.1007/978-3-030-63115-4_3

Elijah AV, Abdullah A, Jhanjhi N, Supramaniam M, and Abdullateef B (2019). Ensemble and deep-learning methods for two-class and multi-attack anomaly intrusion detection: An empirical study. International Journal of Advanced Computer Science and Applications, 10(9): 520-528. https://doi.org/10.14569/IJACSA.2019.0100969

Faker O and Dogdu E (2019). Intrusion detection using big data and deep learning techniques. In the 2019 ACM Southeast Conference, Association for Computing Machinery, Kennesaw, USA: 86-93. https://doi.org/10.1145/3299815.3314439

Feng F, Liu X, Yong B, Zhou R, and Zhou Q (2019). Anomaly detection in ad-hoc networks based on deep learning model: A plug and play device. Ad Hoc Networks, 84: 82-89. https://doi.org/10.1016/j.adhoc.2018.09.014

Gauthama Raman MR, Somu N, Jagarapu S, Manghnani T, Selvam T, Krithivasan K, and Shankar Sriram VS (2020). An efficient intrusion detection technique based on support vector machine and improved binary gravitational search algorithm. Artificial Intelligence Review, 53(5): 3255-3286. https://doi.org/10.1007/s10462-019-09762-z

Gharaee H and Hosseinvand H (2016). A new feature selection IDS based on genetic algorithm and SVM. In the 8th International Symposium on Telecommunications (IST), IEEE, Tehran, Iran: 139-144. https://doi.org/10.1109/ISTEL.2016.7881798

Idhammad M, Afdel K, and Belouch M (2018). Semi-supervised machine learning approach for DDoS detection. Applied Intelligence, 48(10): 3193-3208. https://doi.org/10.1007/s10489-018-1141-2

Kasongo SM and Sun Y (2020). Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset. Journal of Big Data, 7: 105. https://doi.org/10.1186/s40537-020-00379-6

Kshirsagar A and Shah M (2021). Anatomized study of security solutions for multimedia: Deep learning-enabled authentication, cryptography and information hiding. Advanced security solutions for multimedia. In: Ansari IA and Bajaj V (Eds.), Advanced security solutions for multimedia: (7-1)-(7-26). IOP Publishing Ltd., Bristol, UK. https://doi.org/10.1088/978-0-7503-3735-9ch7

Kumar V, Sinha D, Das AK, Pandey SC, and Goswami RT (2020). An integrated rule based intrusion detection system: Analysis on UNSW-NB15 data set and the real time online dataset. Cluster Computing, 23(2): 1397-1418. https://doi.org/10.1007/s10586-019-03008-x

Li G, Yan Z, Fu Y, and Chen H (2018). Data fusion for network intrusion detection: A review. Security and Communication Networks, 2018: 8210614. https://doi.org/10.1155/2018/8210614

Li N, Shepperd M, and Guo Y (2020). A systematic review of unsupervised learning techniques for software defect prediction. Information and Software Technology, 122: 106287. https://doi.org/10.1016/j.infsof.2020.106287

Mabayoje MA, Balogun AO, Ameen AO, and Adeyemo VE (2016). Influence of feature selection on multi-layer perceptron classifier for intrusion detection system. Computing, Information Systems, Development Informatics and Allied Research Journal, 7: 87-94.

Mabayoje MA, Balogun AO, Jibril HA, Atoyebi JO, Mojeed HA, and Adeyemo VE (2019). Parameter tuning in KNN for software defect prediction: An empirical analysis. Jurnal Teknologi dan Sistem Komputer, 7(4): 121-126. https://doi.org/10.14710/jtsiskom.7.4.2019.121-126

Mebawondu JO, Alowolodu OD, Mebawondu JO, and Adetunmbi AO (2020). Network intrusion detection system using supervised learning paradigm. Scientific African, 9: e00497. https://doi.org/10.1016/j.sciaf.2020.e00497

Moustafa N and Slay J (2015). UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In the Military Communications and Information Systems Conference (MilCIS), IEEE, Canberra, Australia: 1-6. https://doi.org/10.1109/MilCIS.2015.7348942 **PMCid:PMC4676426**

Nahar K and Chakraborty P (2020). A modified version of vigenere cipher using 95×95 table. International Journal of Engineering and Advanced Technology (IJEAT), 9(5): 1144-1148. https://doi.org/10.35940/ijeat.E9941.069520

Nawir M, Amir A, Lynn OB, Yaakob N, and Ahmad RB (2018). Performances of machine learning algorithms for binary classification of network anomaly detection system. Journal of Physics: Conference Series: 1st International Conference on Big Data and Cloud Computing, Kuching, Malaysia, 1018: 012015. https://doi.org/10.1088/1742-6596/1018/1/012015

Olasehinde OO (2020). A stacked ensemble intrusion detection approach for the protection of information system. International Journal for Information Security Research, 10: 910-923. https://doi.org/10.20533/ijisr.2042.4639.2020.0105

Saleh AI, Talaat FM, and Labib LM (2019). A hybrid intrusion detection system (HIDS) based on prioritized k-nearest neighbors and optimized SVM classifiers. Artificial Intelligence Review, 51(3): 403-443. https://doi.org/10.1007/s10462-017-9567-1

Salman T, Bhamare D, Erbad A, Jain R, and Samaka M (2017). Machine learning for anomaly detection and categorization in multi-cloud environments. In the IEEE 4th International Conference on Cyber Security and Cloud Computing, IEEE, New York, USA: 97-103. https://doi.org/10.1109/CSCloud.2017.15 **PMid:28017257**

Samoriski JH (2020). Encryption and hacking: Cyphers, hacks and attacks on the digital frontier. In: Filimowicz M and Tzankova V (Eds.), Reimagining communication: Action: 89-106.

Routledge, Milton Park, UK.
https://doi.org/10.4324/9781351015233-5

Saračević MH, Adamović SZ, Miškovic VA, Elhoseny M, Maček ND, Selim MM, and Shankar K (2020). Data encryption for Internet of Things applications based on Catalan objects and two combinatorial structures. IEEE Transactions on Reliability, 70(2): 819-830. https://doi.org/10.1109/TR.2020.3010973

Sarumi OA, Adetunmbi AO, and Adetoye FA (2020). Discovering computer networks intrusion using data analytics and machine intelligence. Scientific African, 9: e00500. https://doi.org/10.1016/j.sciaf.2020.e00500

Sevin A and Mohammed AAO (2021). A survey on software implementation of lightweight block ciphers for IoT devices. Journal of Ambient Intelligence and Humanized Computing: 1-15. https://doi.org/10.1007/s12652-021-03395-3

Sharma J, Giri C, Granmo OC, and Goodwin M (2019). Multi-layer intrusion detection system with ExtraTrees feature selection, extreme learning machine ensemble, and softmax aggregation. EURASIP Journal on Information Security, 2019(1): 1-16. https://doi.org/10.1186/s13635-019-0098-y

Shetty VS, Anusha R, MJ DK, and Hegde P (2020). A survey on performance analysis of block cipher algorithms. In the International Conference on Inventive Computation Technologies, IEEE, Coimbatore, India: 167-174. https://doi.org/10.1109/ICICT48043.2020.9112491

Thaseen IS, Chitturi AK, Al-Turjman F, Shankar A, Ghalib MR, and Abhishek K (2020a). An intelligent ensemble of long-short-term memory with genetic algorithm for network anomaly identification. Transactions on Emerging Telecommunications Technologies, 2020: e4149. https://doi.org/10.1002/ett.4149

Thaseen IS, Poorva B, and Ushasree PS (2020b). Network intrusion detection using machine learning techniques. In the International Conference on Emerging Trends in Information Technology and Engineering, IEEE, Vellore, India: 1-7. https://doi.org/10.1109/ic-ETITE47903.2020.148

Verma P and Shakya M (2021). Machine learning model for predicting major depressive disorder using RNA-Seq data: Optimization of classification approach. Cognitive Neurodynamics: 1-11. https://doi.org/10.1007/s11571-021-09724-8
PMid:35401859

Wei W, Chen S, Lin Q, Ji J, and Chen J (2020). A multi-objective immune algorithm for intrusion feature selection. Applied Soft Computing, 95: 106522. https://doi.org/10.1016/j.asoc.2020.106522

Xin Y, Kong L, Liu Z, Chen Y, Li Y, and Zhu HGM (2018). Machine learning and deep learning methods for cybersecurity. IEEE Access, 6: 35365-35381. https://doi.org/10.1109/ACCESS.2018.2836950