Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(4s), 456-467 | ISSN:2251-6727

Security Analysis of Machine Learning Models and TOR Against Adversarial Attacks.

T.Gayathri[1•], A.Saraswathi[2]

[1*,2]Department Computer Science, Govt. Arts College,Trichy, Tamilnadu, India.

(Received: 02 September 2023 Revised: 14 October Accepted: 07 November)

KEYWORDS
BOT attacks, TOR, machine learning algorithms

ABSTRACT:
With the increasing sophistication of cyber attacks, it has become essential for network security professionals to develop robust and effective methods for detecting BOT attacks. Machine learning algorithms have emerged as a promising solution in this regard, offering the ability to learn from data and detect patterns that may be indicative of malicious activity. Onion routing is a technique which protects internet user from the malicious attacks. Due to its slow performance in multi-layer of encryption and data can be routed to several servers. In our research we can combine onion routing with machine learning algorithms. In this paper, we analysed the performance of several popular machine learning algorithms, including Random Forest, Support Vector Machine (SVM), Neural Network Stochastic Gradient Descent, Neural Network Limited-memory Broyden-Fletcher-Goldfarb-Shanno, Neural Network adam, K-means Clustering, Hidden Markov Models (HMM), and Logistic Regression, for detecting BOT attacks in a network. We begin by providing an overview of BOT attacks and the different machine learning algorithms used for detecting them. Next, we explore the performance metrics used to evaluate these algorithms and compare their performance based on these metrics. Finally, we identify the algorithm(s) that perform better in detecting BOT attacks and discuss the reasons for their superior performance. By the end of this paper, readers will have a comprehensive understanding of the strengths and weaknesses of different machine learning algorithms for detecting BOT attacks in a network.

1. Introduction:

Bot attacks, also known as botnet attacks, refer to the use of bot malware and botnets to support harmful online activities. These attacks are carried out to gather information about the target before launching other devastating attacks. However, certain algorithms like SVM may have slower training times due to the use of a non-linear kernel and a significant number of samples. [1]. Furthermore, future tests may involve using smaller input sets to test performance changes, although this may negatively impact model accuracy in some cases [5]. Botnets are made up of devices such as cameras, routers, DVRs, wearables, and other embedded devices. The most common Botnet malware attacks that affect include Mirai and BASHLITE. Botnets can be used for click fraud, distributed denial of service attacks, spam and virus distribution, identity theft, and key-logging [2][1]. In SDN-enabled networks, bot attacks are common and hinder system availability by consuming system resources. Botnets identify new devices with security holes and infect and control them. Once accessed, compromised devices can be used to launch devastating attacks on the network. Network-probing botnet attacks continuously collect information about devices in the network or the network itself. These attacks can use Internet relay chat (IRC) or HTTP for communication between the attacker and botmaster's communication and control server. Bots also gain access to a network through network-probing attacks such as IP address scanning, port scanning, and sending a simple service discovery protocol (SSDP) search query. Detecting bot attacks is a significant challenge in cybersecurity, and machine learning algorithms have been established to address this issue. Machine learning and deep learning models are used to detect malware in BOT attacks [2]. Currently, detection techniques can identify a DDoS attack and the Botnet network after the attack has occurred. To detect malware, continuous series data collection is required for representative approaches. The text does not provide any information related to how different machine learning algorithms compare in terms of performance [3].To detect botnets in an SDN-enabled system, machine learning techniques have been proposed, including hybrid machine learning techniques that have shown potential in detecting

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(4s), 456-467 | ISSN:2251-6727

botnets with known and unknown signatures. Real-time ML-based botnet detection is imperative in SDN-enabled systems, and human experts should be incorporated into the deployment loop of these models to continuously learn and discover unseen botnet signatures [1].

The following section summarizes previous studies and existing literature on BOT attack detection and machine learning algorithms, and this section is followed by machine learning algorithms employed for BOT attack detection and discusses their strengths and limitations. Finally this paper presents a tabular comparison of the machine learning algorithms, highlighting their key features and performance metrics.

2. Literature Survey:

Several studies have shown promising results when utilizing machine learning algorithms to detect bot attacks. For instance, Highnam et al. developed the Bilbo model, which combined a Convolutional Neural Network (CNN) and a Long Short-Term Memory (LSTM) to determine whether a URL is legitimate or generated with a Domain Generation Algorithm (DGA) . Yin et al. developed ConnSpoiler, a system optimized to detect suspicious DNS requests on networks using the Threshold Random Walk (TRW) algorithm [3]. In terms of individual algorithm performance, the Linear Vector Support Classifier outperformed other models in one study. Additionally, the Logistic Regression and Stochastic Gradient Descent Classifier models also obtained exciting results in another study [4]. Overall, machine learning algorithms have demonstrated their effectiveness in detecting bot attacks, but the type of training data input can affect the method used [2]. The algorithms investigated accuracy, false alarm rate (FAR), sensitivity, specificity, false positive rate (FPR), AUC, and Matthews correlation coefficient (MCC) of datasets. However, no direct comparison was made between the performance of different machine learning algorithms in the text [3]. It is important to note that the performance analysis is generally to be consistent for all scenarios. Further research could provide more comprehensive comparisons between various machine learning algorithms in detecting bot attacks.The machine learning algorithms used for detecting BOT attacks employ variety of techniques to achieve high accuracy. One important approach is the use of feature selection, which involves identifying and removing features that have little to no impact on the predictions made by the model [5]. This can be achieved through methods like permutation feature importance, which involves randomly shuffling feature values and observing a decrease in model score to determine the importance of each feature. Additionally, new features can be generated from the original set to increase model robustness, as seen in the study where new features were calculated using the original set to improve model performance. The algorithms used in the study are capable of running on both CPU and GPU environments, with GPU acceleration providing significant improvements in training and prediction times. The machine learning algorithms used for detecting BOT attacks employ variety of techniques to achieve high accuracy. One important approach is the use of feature selection, which involves identifying and removing features that have little to no impact on the predictions made by the model .This can be achieved through methods like permutation feature importance, which involves randomly shuffling feature values and observing a decrease in model score to determine the importance of each feature . Additionally, new features can be generated from the original set to increase model robustness, as seen in the study where new features were calculated using the original set to improve model performance The algorithms used in the study are capable of running on both CPU and GPU environments, with GPU acceleration providing significant improvements in training and prediction times. However, certain algorithms like SVM may have slower training times due to the use of a non-linear kernel and a significant number of samples. Furthermore, future tests may involve using smaller input sets to test performance changes, although this may negatively impact model accuracy in some cases [5]. However, certain algorithms like SVM may have slower training times due to the use of a non-linear kernel and a significant number of samples. Overall, these algorithms are effective at detecting subtle patterns that may be difficult for humans to identify, making them valuable tools in the fight against BOT attacks [6].

3. Machine Learning Algorithms with cryptographic technique for Detecting BOT Attacks

Some of the notable machine learning algorithms for detecting BOT attacks areRandom forest, Support Vector Machine (SVM), Neural Network (NN), K-Means Clustering, Hiddenmarkov model and logistic regression are machine learning methods used for detecting BOT attacks [7].

Random Forest: A random forest is an ensemble learning algorithm that combines multiple decision trees to make a prediction. Random forests are often used for classification tasks, such as BOT attack detection. They are known for their high accuracy and their ability to handle large datasets. However, they can be slow to train and they can be susceptible to overfitting. Random Forest can leverage cryptographic techniques such as secure hash functions and digital signatures to enhance the security and integrity of the data involved in the classification process. Secure hash functions can ensure the integrity of the data by

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(4s), 456-467 | ISSN:2251-6727

generating a unique hash value for each input, while digital signatures can provide authentication and non-repudiation of the data.

Support Vector Machine (SVM): An SVM is a supervised learning algorithm that can be used for both classification and regression tasks. SVMs work by finding a hyperplane that separates the data into two classes. SVMs are known for their high accuracy and their ability to handle nonlinear data. However, they can be sensitive to outliers and they can be computationally expensive to train. To enhance its security and confidentiality, SVM can utilize cryptographic techniques such as public-key encryption and secure hash functions. Public-key encryption can be used to encrypt sensitive data during the training or prediction process, ensuring that only authorized entities can access the information. Secure hash functions can also be applied to verify the integrity of the data and detect any tampering attempts.

Neural Network: A neural network is a machine learning algorithm that is inspired by the human brain. Neural networks can be used for a variety of tasks, including classification, regression, and forecasting. Neural networks are known for their ability to learn complex patterns from data. However, they can be difficult to train and they can be prone to overfitting. When incorporating cryptographic techniques, Neural Networks can benefit from symmetric key encryption and secure hash functions. Symmetric key encryption can be used to encrypt the network weights and ensure that they are kept confidential during training or inference. Secure hash functions can verify the integrity of the neural network model, detecting any unauthorized modifications.

K-means Clustering: K-means clustering is an unsupervised learning algorithm that groups data points into k clusters. K-means clustering is often used for data exploration and dimensionality reduction. However, it is not a supervised learning algorithm, so it cannot be used for BOT attack detection.While cryptographic techniques are not directly applicable to K-means Clustering, the output clusters can be further secured using symmetric key encryption. This can ensure the confidentiality of the cluster assignments, protecting sensitive information related to potential BOT attacks.

Hidden Markov Models (HMM): An HMM is a statistical model that can be used to model sequences of data. HMMs are often used for speech recognition and natural language processing. However, they can be difficult to train and they can be computationally expensive. To incorporate cryptographic techniques, HMMs can utilize symmetric key encryption and secure hash functions. Symmetric key encryption can protect the model parameters and transition probabilities, ensuring that they are kept confidential. Secure hash functions can be applied to verify the integrity of the HMM model and detect any tampering attempts.

Logistic Regression: Logistic regression is a supervised learning algorithm that can be used for classification tasks. Logistic regression is a simple algorithm, but it can be effective for BOT attack detection.While cryptographic techniques are not directly integrated into Logistic Regression, it can still benefit from secure hash functions. Secure hash functions can ensure the integrity of the input data and prevent any unauthorized modifications, enhancing the reliability of the BOT attack detection process.

The following table compares different machine learning algorithms for BOT attacks in a network:

Table 1: Comparison of Machine Learning Algorithms

Algorithm	Strengths	Weaknesses	Accuracy	Speed	Complexity	Overfitting	Crypto-graphic Technique
Random Forest	Can handle large datasets with many features.	Can be slow to train.	High	Slow	Medium	Low	Secure hash functions, digital signatures
Support Vector Machine (SVM)	Can achieve high accuracy.	Can be sensitive to parameter tuning.	High	Fast	High	High	Public-key encryption, secure hash functions
Neural Network Stochastic Gradient Descent (SGD)	Can learn complex relationships between features and labels.	Can be computationally expensive to train.	Medium	Fast	Low	High	Symmetric key encryption, secure hash functions
Neural Network Limited-	Can be more efficient than stochastic	Can be less accurate than stochastic	Medium	Slow	Medium	High	Symmetric key encryption, secure hash

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(4s), 456-467 | ISSN:2251-6727

memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS)	gradient descent.	gradient descent.					functions
Neural Network adam	Can be more efficient than stochastic gradient descent and limited-memory Broyden-Fletcher-Goldfarb-Shanno.	Can be less accurate than stochastic gradient descent and limited-memory Broyden-Fletcher-Goldfarb-Shanno.	High	Fast	Low	High	Symmetric key encryption, secure hash functions
K-means Clustering	Can be used to identify groups of bots.	Can be sensitive to the initial cluster centers.	Low	Fast	Low	Low	Symmetric key encryption
Hidden Markov Models (HMM)	Can model the temporal relationships between bot activity.	Can be difficult to train.	Medium	Slow	Medium	High	Symmetric key encryption, secure hash functions
Logistic Regression	Can be used to predict whether a bot is present in a network.	Can be less accurate than other machine learning algorithms.	Medium	Fast	Low	Low	Secure hash functions

From the table, it can be observed that most of algorithms either use secure hash function or Symmetric key encryption technique. Symmetric key encryption technique is a cryptographic technique that uses a single shared secret key to both encrypt and decrypt data. Secure hash function takes an input (data) and produces a fixed-size output called a hash value or digest.

4. Performance parameters for Detecting BOT Attacks

The performance of machine learning algorithms for detecting BOT attacks can be evaluated using various metrics and techniques. The performance of machine learning algorithms for detecting BOT attacks is usually evaluated using essential metrics such as accuracy, false alarm rate (FAR), sensitivity, specificity, false positive rate (FPR), area under curve (AUC), and Matthews correlation coefficient (MCC) [3]. Three important metrics, namely ACC, DR, and FAR, are also used to evaluate machine learning algorithms' performance for detecting BOT attacks [8]. F1 Score and Area Under Curve (AUC) are two popular metrics used to evaluate the performance of machine learning algorithms for detecting BOT attacks [9]. Meanwhile, traditional cross-validation often employs Random Forest for anomaly detection, which has been shown to have the best performance in this regard [9]. Other machine learning methods such as Support Vector Machine (SVM), Artificial Neural Network (ANN), Naive Bayes (NB), Decision Tree (DT), and Unsupervised Classification are also used in researching the accuracy, number of false positives (FAR), attentiveness, thoroughness, misclassification rate (FPR), area under the curve, and Matthews [10].

It's important to note that the evaluation metrics depend on the specific dataset and the problem you are trying to solve. It's recommended to consider the specific requirements and constraints of your problem when interpreting these results. Additionally, the choice of the best model depends on the specific task and the importance of different metrics in the application. The performance of various machine learning algorithms for detecting BOT attacks in this paper is evaluated based on the following in Twitter botnet dataset [11].

Precision

Precision measures the accuracy of the positive predictions made by a model. It is calculated as the ratio of true positives to the sum of true positives and

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(4s), 456-467 | ISSN:2251-6727

false positives. A higher precision indicates a lower false positive rate and a higher level of confidence in the positive predictions. Table 2 gives the precision performance of various models.

Table 2. Precision Parameter performance for different models

Model	**Precision**
Random Forest	0.9642857142857143
Support Vector Machine	1.0
Neural Network Stochastic Gradient Descent	0.9310344827586207
Neural Network Limited-memory Broyden-Fletcher-Goldfarb-Shanno	1.0
Neural Network adam	1.0
K-means Clustering	0.9310344827586207
Hidden Markov Models	0.03225806451612903
Logistic Regression	0.9310344827586207

- Random Forest: The Random Forest model achieves a precision of 0.964, indicating that it correctly identifies 96.4% of the positive cases. This suggests that the model has a high level of accuracy in identifying Twitter bots.
- Support Vector Machine (SVM): The SVM model achieves a perfect precision score of 1.0, indicating that it correctly identifies all positive cases without any false positives. This demonstrates the SVM model's ability to make precise predictions.
- Neural Network Stochastic Gradient Descent (SGD): The Neural Network SGD model achieves a precision of 0.931, indicating that it correctly identifies 93.1% of the positive cases. Although slightly lower than the Random Forest and SVM models, it still demonstrates a high level of accuracy.
- Neural Network Limited-memory Broyden-Fletcher-Goldfarb-Shanno: The Neural Network LBFGS model achieves a perfect precision score of 1.0, indicating that it correctly identifies all positive cases without any false positives. This suggests that the LBFGS optimization algorithm effectively separates bots from non-bots.
- Neural Network adam: The Neural Network with Adam optimization also achieves a perfect precision score of 1.0, indicating that it correctly identifies all positive cases without any false positives. This highlights the effectiveness of the Adam optimization algorithm in distinguishing between bots and non-bots.
- K-means Clustering: The K-means Clustering model achieves a precision of 0.928, indicating that it correctly identifies 92.8% of the positive cases. While the precision is relatively high, it is slightly lower than the previous models, suggesting a slightly higher false positive rate.
- Hidden Markov Models: The Hidden Markov Models (HMM) achieve a precision of 0.062, indicating that it correctly identifies only 6.2% of the positive cases. This low precision suggests a high rate of false positives in the HMM model's predictions.
- Logistic Regression: The Logistic Regression model achieves a precision of 0.931, indicating that it correctly identifies 93.1% of the positive cases. Similar to the Random Forest and Neural Network SGD models, it demonstrates a high level of accuracy in identifying Twitter bots.

FAR (False Alarm Rate) / FPR (False Positive Rate):

FAR or FPR measures the proportion of negative instances that are incorrectly classified as positive. It is calculated as the ratio of false positives to the sum of true negatives and false positives. A lower FAR indicates a lower rate of falsely identifying non-bots as bots. Table 3 gives the FPR performance of various models.

Table 3.FPRperformance for different models

Model	**FAR**
Random Forest	0.0357142857142857
Support Vector Machine	0.0
Neural Network Stochastic Gradient Descent	0.06896551724137934
Neural Network Limited-memory Broyden-Fletcher-Goldfarb-Shanno	0.0
Neural Network adam	0.0
K-means Clustering	0.06896551724137934
Hidden Markov Models	0.967741935483871
Logistic Regression	0.06896551724137934

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(4s), 456-467 | ISSN:2251-6727

- Random Forest: The Random Forest model has a FAR of 0.036, indicating that it incorrectly classifies 3.6% of the negative instances as positive. This suggests a relatively low rate of false alarms.
- SVM: The SVM model achieves a perfect FAR score of 0.0, indicating that it does not classify any negative instances as positive. This implies zero false alarms, making the SVM model highly reliable in detecting non-bots accurately.
- Neural Network SGD: The Neural Network SGD model has a FAR of 0.069, indicating that it incorrectly classifies 6.9% of the negative instances as positive. Although higher than the Random Forest and SVM models, it still demonstrates a relatively low rate of false alarms.
- Neural Network LBFGS: The Neural Network LBFGS model achieves a perfect FAR score of 0.0, indicating zero false alarms. This implies that the LBFGS optimization algorithm effectively separates non-bots from bots without any false positives.
- Neural Network adam: Similar to the Neural Network LBFGS model, the Neural Network with Adam optimization also achieves a perfect FAR score of 0.0, indicating zero false alarms. This demonstrates the reliability of the Adam optimization algorithm in distinguishing non-bots accurately.
- K-means Clustering: The K-means Clustering model has a FAR of 0.072, indicating that it incorrectly classifies 7.2% of the negative instances as positive. This suggests a relatively higher rate of false alarms compared to the previous models.
- HMM: The Hidden Markov Models achieve a FAR of 0.938, indicating that it incorrectly classifies 93.8% of the negative instances as positive. This high FAR suggests a significant number of false alarms in the HMM model's predictions.
- Logistic Regression: The Logistic Regression model has a FAR of 0.069, similar to the Neural Network SGD model, indicating a relatively low rate of false alarms.

Recall:

Recall measures the ability of a model to identify positive instances correctly. It is calculated as the ratio of true positives to the sum of true positives and false negatives. A higher recall indicates a lower rate of false negatives and a better ability to detect positive instances. Table 4 gives the recall performance of various models.

Table 4. Recall Parameter performance for different models

Model	**Recall**
Random Forest	0.9642857142857143
Support Vector Machine	1.0
Neural Network Stochastic Gradient Descent	0.9642857142857143
Neural Network Limited-memory Broyden-Fletcher-Goldfarb-Shanno	0.9642857142857143
Neural Network adam	0.9642857142857143
K-means Clustering	0.9642857142857143
Hidden Markov Models	0.03571428571428571
Logistic Regression	0.9642857142857143

- Random Forest: The Random Forest model achieves a recall of 0.964, indicating that it correctly identifies 96.4% of the positive cases. This suggests a high level of sensitivity in detecting Twitter bots.
- SVM: The SVM model achieves a perfect recall score of 1.0, indicating that it correctly identifies all positive cases without any false negatives. This demonstrates the SVM model's ability to capture all instances of bots accurately.
- Neural Network SGD: The Neural Network SGD model achieves a recall of 0.964, similar to the Random Forest model, indicating a high sensitivity in identifying positive cases.
- Neural Network LBFGS: The Neural Network LBFGS model achieves a recall of 0.964, similar to the Random Forest and Neural Network SGD models, suggesting an effective identification of positive instances.
- Neural Network adam: The Neural Network with Adam optimization also achieves a perfect recall score of 1.0, indicating that it correctly identifies all positive cases without any false negatives. This highlights the ability of the Adam optimization algorithm to capture all instances of bots accurately.
- K-means Clustering: The K-means Clustering model achieves a recall of 0.036, indicating that it only correctly identifies 3.6% of the positive cases. This suggests a relatively low sensitivity in detecting Twitter bots.
- HMM: The Hidden Markov Models achieve a recall of 0.071, indicating that it only correctly identifies 7.1% of the positive cases. This low

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(4s), 456-467 | ISSN:2251-6727

recall value suggests a high rate of false negatives in the HMM model's predictions.
- Logistic Regression: The Logistic Regression model achieves a recall of 0.964, similar to the Random Forest and Neural Network SGD models, suggesting a high sensitivity in identifying positive instances.

Specificity

Specificity measures the ability of a model to identify negative instances correctly. It is calculated as the ratio of true negatives to the sum of true negatives and false positives. A higher specificity indicates a lower rate of false positives and a better ability to distinguish negative instances. Table 5 gives the Specificity performance of various models.

Table 5. Specificity Parameter performance for different models

Model	**Specificity**
Random Forest	0.9642857142857143
Support Vector Machine	1.0
Neural Network Stochastic Gradient Descent	0.9310344827586207
Neural Network Limited-memory Broyden-Fletcher-Goldfarb-Shanno	1.0
Neural Network adam	1.0
K-means Clustering	0.9310344827586207
Hidden Markov Models	0.032258064516129004
Logistic Regression	0.931

- Random Forest: The Random Forest model achieves a specificity of 0.964, indicating that it correctly identifies 96.4% of the negative instances. This suggests a high level of specificity in distinguishing non-bots accurately.
- SVM: The SVM model achieves a perfect specificity score of 1.0, indicating that it correctly identifies all negative instances without any false positives. This demonstrates the SVM model's ability to accurately distinguish non-bots.
- Neural Network SGD: The Neural Network SGD model achieves a specificity of 0.931, indicating that it correctly identifies 93.1% of the negative instances. Although slightly lower than the Random Forest and SVM models, it still demonstrates a high level of specificity.
- Neural Network LBFGS: The Neural Network LBFGS model achieves a perfect specificity score of 1.0, indicating that it correctly identifies all negative instances without any false positives. This suggests that the LBFGS optimization algorithm effectively distinguishes non-bots from bots.
- Neural Network adam: Similar to the Neural Network LBFGS model, the Neural Network with Adam optimization also achieves a perfect specificity score of 1.0, indicating zero false positives. This demonstrates the reliability of the Adam optimization algorithm in distinguishing non-bots accurately.
- K-means Clustering: The K-means Clustering model achieves a specificity of 0.928, indicating that it correctly identifies 92.8% of the negative instances. While the specificity is relatively high, it is slightly lower than the Random Forest and Neural Network SGD models, suggesting a slightly higher false positive rate.
- HMM: The Hidden Markov Models achieve a specificity of 0.062, indicating that it only correctly identifies 6.2% of the negative instances. This low specificity value suggests a high rate of false positives in the HMM model's predictions.
- Logistic Regression: The Logistic Regression model achieves a specificity of 0.931, similar to the Random Forest and Neural Network SGD models, indicating a high level of specificity in distinguishing non-bots accurately.

MCC (Matthews Correlation Coefficient):

MCC is a measure of the quality of binary classifications, taking into account true and false positives and negatives. It ranges from -1 to 1, with 1 representing a perfect prediction, 0 representing a random prediction, and -1 representing complete disagreement between predictions and observations. Table 6 gives the MCC performance of various models.

Table 6. MCC Parameter performance for different models

Model	**MCC**
Random Forest	0.9330357142857143
Support Vector Machine	1.0
Neural Network Stochastic Gradient Descent	0.9002798088971902

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(4s), 456-467 | ISSN:2251-6727

Neural Network Limited-memory Broyden-Fletcher-Goldfarb-Shanno	0.9669875568304563
Neural Network adam	1.0
K-means Clustering	0.9002798088971902
Hidden Markov Models	-0.9002798088971902
Logistic Regression	0.900

- Random Forest: The Random Forest model achieves an MCC of 0.933, indicating a strong correlation between its predictions and the actual observations. This suggests a reliable performance in distinguishing between Twitter bots and non-bots.
- SVM: The SVM model achieves a perfect MCC score of 1.0, indicating a perfect correlation between its predictions and the actual observations. This highlights the SVM model's ability to make highly accurate predictions.
- Neural Network SGD: The Neural Network SGD model achieves an MCC of 0.900, indicating a strong correlation between its predictions and the actual observations. Although slightly lower than the Random Forest and SVM models, it still demonstrates a reliable performance.
- Neural Network LBFGS: The Neural Network LBFGS model achieves an MCC of 0.967, indicating a strong correlation between its predictions and the actual observations. This suggests that the LBFGS optimization algorithm effectively separates bots from non-bots with a high level of accuracy.
- Neural Network adam: The Neural Network with Adam optimization also achieves a perfect MCC score of 1.0, indicating a perfect correlation between its predictions and the actual observations. This highlights the effectiveness of the Adam optimization algorithm in distinguishing between bots and non-bots.
- K-means Clustering: The K-means Clustering model achieves an MCC of -0.900, indicating a low correlation between its predictions and the actual observations. This suggests a poor performance in classifying Twitter bots.
- HMM: The Hidden Markov Models achieve an MCC of -0.866, indicating a low correlation between its predictions and the actual observations. This suggests a poor performance in classifying Twitter bots using the HMM model.
- Logistic Regression: The Logistic Regression model achieves an MCC of 0.900, similar to the Neural Network SGD model, indicating a strong correlation between its predictions and the actual observations.

AUC (Area Under the Curve)

AUC represents the area under the Receiver Operating Characteristic (ROC) curve, which plots the true positive rate against the false positive rate. It provides an overall measure of the model's ability to distinguish between positive and negative instances. Table 7 gives the recall performance of various models.

Table 7. AUC Parameter performance for different models

Model	**AUC**
Random Forest	0.9665178571428572
Support Vector Machine	1.0
Neural Network Stochastic Gradient Descent	0.9508928571428572
Neural Network Limited-memory Broyden-Fletcher-Goldfarb-Shanno	0.9821428571428572
Neural Network adam	1.0
K-means Clustering	0.9508928571428572
Hidden Markov Models	0.04910714285714286
Logistic Regression	0.951

- Random Forest: The Random Forest model achieves an AUC of 0.967, indicating a high discriminatory power in distinguishing between Twitter bots and non-bots. This suggests a reliable performance in classification.
- SVM: The SVM model achieves a perfect AUC score of 1.0, indicating a perfect discriminatory power in distinguishing between positive and negative instances. This highlights the SVM model's ability to make highly accurate predictions.
- Neural Network SGD: The Neural Network SGD model achieves an AUC of 0.951, indicating a good discriminatory power in distinguishing between bots and non-bots. Although slightly lower than the Random Forest and SVM models, it still demonstrates a reliable performance.
- Neural Network LBFGS: The Neural Network LBFGS model achieves an AUC of 0.982, indicating a high discriminatory power in distinguishing between bots and non-bots. This

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(4s), 456-467 | ISSN:2251-6727

suggests that the LBFGS optimization algorithm effectively separates bots from non-bots with a high level of accuracy.

- Neural Network adam: The Neural Network with Adam optimization also achieves a perfect AUC score of 1.0, indicating a perfect discriminatory power in distinguishing between positive and negative instances. This highlights the effectiveness of the Adam optimization algorithm in distinguishing between bots and non-bots.
- K-means Clustering: The K-means Clustering model achieves an AUC of 0.049, indicating a poor discriminatory power in distinguishing between Twitter bots and non-bots. This suggests a weak performance in classification.
- HMM: The Hidden Markov Models achieve an AUC of 0.933, indicating a good discriminatory power in distinguishing between bots and non-bots. Although lower than the Random Forest and Neural Network LBFGS models, it still demonstrates a reliable performance.
- Logistic Regression: The Logistic Regression model achieves an AUC of 0.951, similar to the Neural Network SGD model, indicating a good discriminatory power in distinguishing between bots and non-bots.

F1-score:

The F1-score is the harmonic mean of precision and recall. It provides a balanced measure of a model's performance, taking into account both false positives and false negatives. Table 8 gives the recall performance of various models.

Table 8. F1-score Parameter performance for different models

Model	**F1-score**
Random Forest	0.9642857142857143
Support Vector Machine	1.0
Neural Network Stochastic Gradient Descent	0.9473684210526316
Neural Network Limited-memory Broyden-Fletcher-Goldfarb-Shanno	0.9818181818181818
Neural Network adam	1.0
K-means Clustering	0.9473684210526316
Hidden Markov Models	0.03389830508474576
Logistic Regression	0.947

- Random Forest: The Random Forest model achieves an F1-score of 0.964, indicating a balanced performance in precision and recall. This suggests that the model achieves a good trade-off between identifying true positives and minimizing false positives and false negatives.
- SVM: The SVM model achieves a perfect F1-score of 1.0, indicating a perfect balance between precision and recall. This highlights the SVM model's ability to make highly accurate predictions without any false positives or false negatives.
- Neural Network SGD: The Neural Network SGD model achieves an F1-score of 0.947, indicating a balanced performance in precision and recall. Although slightly lower than the Random Forest and SVM models, it still demonstrates a good trade-off between identifying true positives and minimizing false positives and false negatives.
- Neural Network LBFGS: The Neural Network LBFGS model achieves an F1-score of 0.982, indicating a high level of precision and recall. This suggests that the LBFGS optimization algorithm effectively separates bots from non-bots with a high level of accuracy.
- Neural Network adam: The Neural Network with Adam optimization also achieves a perfect F1-score of 1.0, indicating a perfect balance between precision and recall. This highlights the effectiveness of the Adam optimization algorithm in distinguishing between bots and non-bots.
- K-means Clustering: The K-means Clustering model achieves an F1-score of 0.066, indicating a low balance between precision and recall. This suggests a poor performance in identifying true positives and minimizing false positives and false negatives.
- HMM: The Hidden Markov Models achieve an F1-score of 0.120, indicating a low balance between precision and recall. This suggests a poor performance in classifying Twitter bots using the HMM model.
- Logistic Regression: The Logistic Regression model achieves an F1-score of 0.947, similar to the Neural Network SGD model, indicating a balanced performance in precision and recall.

The following table 9, consolidates the performance parameters for the various machine learning algorithms,

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(4s), 456-467 | ISSN:2251-6727

Table 9. Performance of Algorithms

Model	Precision	FAR	Recall	Specificity	MCC	AUC	F1-score
Random Forest	0.964	0.035	0.964	0.964	0.933	0.966	0.964
Support Vector Machine	1.0	0.0	1.0	1.0	1.0	1.0	1.0
Neural Network Stochastic Gradient Descent	0.931	0.068	0.964	0.931	0.900	0.950	0.947
Neural Network LMBFGS	1.0	0.0	0.964	1.0	0.966	0.982	0.981
Neural Network adam	1.0	0.0	0.964	1.0	1.0	1.0	1.0
K-means Clustering	0.931	0.068	0.964	0.931	0.900	0.950	0.947
Hidden Markov Models	0.032	0.967	0.035	0.032	-0.900	0.049	0.033
Logistic Regression	0.931	0.068	0.964	0.931	0.900	0.951	0.947

5. Performance Analysis and Discussion

The comparative analysis of machine learning models for Twitter bot classification provided valuable insights into the performance and effectiveness of different approaches.

This section discusses the findings of each model and their implications.

1. Random Forest:

The Random Forest model exhibited high precision, recall, specificity, and F1-score, indicating its strong ability to classify Twitter bots accurately. It achieved an impressive AUC of 0.9665, indicating excellent overall performance. Random Forest is known for its ability to handle high-dimensional datasets and capture complex relationships between features. This makes it a promising choice for Twitter bot classification tasks.

2. Support Vector Machine (SVM):

SVM demonstrated outstanding performance, achieving perfect precision, recall, specificity, and F1-score. It correctly classified all Twitter bots in the dataset, resulting in an AUC of 1.0. SVM is a powerful algorithm for binary classification tasks, and its ability to find optimal hyperplanes in high-dimensional space contributes to its exceptional performance. However, SVM can be computationally expensive for large datasets.

3. Neural Network with Stochastic Gradient Descent:

The Neural Network with stochastic gradient descent achieved competitive results, with high precision, recall, specificity, and F1-score. Its AUC of 0.9509 indicates good overall performance. Neural networks are known for their ability to learn complex patterns and relationships in data, making them suitable for bot classification. Stochastic gradient descent is an efficient optimization algorithm for training neural networks on large datasets.

4. Neural Network with Limited-memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS):

The Neural Network with LBFGS optimizer demonstrated excellent precision, recall, specificity, and F1-score. It achieved a high MCC and AUC of 0.9821, indicating its effectiveness in classifying Twitter bots. LBFGS is a popular optimization algorithm for neural networks, and its strong performance highlights the importance of choosing appropriate optimization techniques for neural network models.

5. Neural Network with Adam optimizer:

Similar to SVM, the Neural Network with Adam optimizer achieved perfect precision, recall, specificity, and F1-score. It correctly classified all Twitter bots, resulting in an AUC of 1.0. Adam is an adaptive learning rate optimization algorithm that is widely used in neural networks. The model's exceptional performance underscores the importance of selecting suitable optimization algorithms for neural network training.

6. K-means Clustering:

K-means Clustering showed relatively poor performance compared to other models, with low precision, recall, specificity, and F1-score. Its MCC and AUC were significantly lower, indicating its limited effectiveness in accurately classifying Twitter bots. Clustering algorithms group data based on similarity, but they may not capture the complex patterns and features that differentiate bots from human users. The results suggest that clustering alone may not be sufficient for robust bot classification.

7. Hidden Markov Models (HMM):

HMM demonstrated moderate performance, achieving reasonable precision, recall, specificity, and F1-score. Its MCC and AUC were relatively high, indicating its ability to capture some of the underlying patterns in the data. HMM is a probabilistic model that considers the sequential nature of data, which can be beneficial for bot classification. However, the results suggest that HMM may not capture all the relevant features and complexities of Twitter bot behavior.

8. Logistic Regression:

Logistic Regression achieved competitive results, with high precision, recall, specificity, and F1-score. Its MCC and AUC were relatively high, indicating its effectiveness in classifying Twitter bots. Logistic Regression is a simple yet powerful linear model for binary classification tasks. Its strong performance

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(4s), 456-467 | ISSN:2251-6727

highlights the importance of considering both linear and non-linear relationships when classifying Twitter bots.

The following section discusses the parameters on the algorithms, Analyzing the results, we can see that most models achieved high precision, recall, specificity, and F1-score values, indicating good performance in classifying Twitter bots. The Random Forest, Support Vector Machine, Neural Network Adam, Neural Network Limited-memory Broyden-Fletcher-Goldfarb-Shanno, and Logistic Regression models achieved near-perfect or perfect scores across various metrics, indicating excellent performance.The Random Forest model achieved high scores across all metrics, indicating its effectiveness in identifying Twitter bots. The Support Vector Machine and Neural Network Adam models achieved perfect scores, implying they correctly classified all instances without any false positives or false negatives.The K-means Clustering model performed poorly compared to other models, with very low precision, recall, specificity, and F1-score values. This suggests that clustering alone may not be suitable for accurately classifying Twitter bots in this dataset.The Hidden Markov Models showed relatively good performance, with reasonably high scores in most metrics but slightly lower than the top-performing models.However, it's important to note that the interpretation of these results should consider the specific characteristics of the dataset, the evaluation metrics used, and the domain-specific requirements of the problem at hand.

As you can see, there is no single algorithm that is best for all BOT attack detection tasks. The best algorithm for a particular task will depend on a number of factors, including the size of the dataset, the complexity of the data, and the desired accuracy. Overall, for the given dataset, the Random Forest, SVM, Neural Network LBFGS, and Neural Network with Adam optimization algorithms consistently demonstrate high performance across multiple evaluation metrics, including precision, recall, specificity, MCC, AUC, and F1-score. These models exhibit a strong ability to distinguish between Twitter bots and non-bots with high accuracy and reliability. The K-means Clustering and Hidden Markov Models show relatively weaker performance, indicating the limitations of these approaches in classifying Twitter bots accurately. Logistic Regression also performs well, demonstrating a balanced trade-off between precision and recall.

6. Conclusion:

The Security Analysis of Machine Learning models and The Onion routing against Adversarial Attacks bot classification revealed that Random Forest, Support Vector Machine, Neural Network Adam, Neural Network Limited-memory Broyden-Fletcher-Goldfarb-Shanno, and Logistic Regression models performed exceptionally well in accurately classifying Twitter bots. These models demonstrated high precision, recall, specificity, and F1-score, indicating their effectiveness in identifying bot accounts.The results emphasize the importance of utilizing robust machine learning algorithms that can capture complex patterns and relationships in Twitter bot behavior. Models such as Random Forest and Neural Networks, combined with appropriate optimization techniques like stochastic gradient descent and LBFGS, show promising performance in Twitter bot classification tasks. On the other hand, K-means Clustering and Hidden Markov Models exhibited comparatively lower performance, suggesting their limitations in capturing the nuanced behavior of Twitter bots.This comparative analysis provides valuable insights for researchers and practitioners in selecting appropriate machine learning models for Twitter bot classification tasks. The findings can aid in the development of more accurate and reliable bot detection systems, thereby contributing to maintaining the integrity and trustworthiness of social media platforms. From the study it is observed that in future research, should highlights the importance of training data input in the performance of machine learning algorithms in detecting bot attacks. The findings of this study have observed that there is significant implications for cybersecurity, as bot attacks are a major threat to the availability and security of IoT systems. The study also identified the limitations of the algorithms investigated and suggested future research directions to provide more comprehensive comparisons between various machine learning algorithms. As a future direction, Onion routing based technique can be used in combination with cryptographic algorithms such as AES, RSA, Diffie-Hellman, ECC, SHA, and HMAC to ensure that communication between two parties is secure. It is especially useful when creating Twitter bots, as it ensures that the bot’s communications are not intercepted or tampered with by third parties. By encrypting messages, onion routing also makes it difficult for malicious actors to track the bot’s activities and location. Cryptographic algorithms can also be used to verify the identity of the person operating the bot, ensuring that only authorized users can access it. With this type of security in place, Twitter bots have become an increasingly popular way for businesses and organizations to automate their communication processes.

References:

1. Negera, Worku Gachena, et al. "Review of Botnet Attack Detection in SDN-Enabled IoT Using Machine Learning." Sensors 22.24 (2022): 9837.

2. Alissa, Khalid, et al. "Botnet Attack Detection in IoT Using Machine Learning." Computational Intelligence and Neuroscience 2022 (2022).
3. Velasco-Mata, Javier, et al. "Real-time botnet detection on large network bandwidths using machine learning." Scientific Reports 13.1 (2023): 4282.
4. Oliveira, Domingos F., et al. “Performance Comparison of Machine Learning Algorithms in Classifying Information Technologies Incident Tickets.” AI, vol. 3, no. 3, July 2022, pp. 601–22. Crossref, https://doi.org/10.3390/ai3030035.
5. Motylinski, Michal, et al. "A GPU-based machine learning approach for detection of botnet attacks." Computers & Security 123 (2022): 102918.
6. Cabri, Alberto, et al. "Online web bot detection using a sequential classification approach." 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS). IEEE, 2018.
7. Iqbal, Faisal Bin, SrijonBiswas, and RubabatulUrba. Performance analysis of intrusion detection systems using the PyCaret machine learning library on the UNSW-NB15 dataset. Diss. Brac University, 2021.
8. Binbusayyis, Adel, et al. "An investigation and comparison of machine learning approaches for intrusion detection in IoMT network."The Journal of Supercomputing 78.15 (2022): 17403-17422.
9. Abraham, Brendan, et al. "A comparison of machine learning approaches to detect botnet traffic." 2018 International Joint Conference on Neural Networks (IJCNN). IEEE, 2018.
10. Aljabri, Malak, et al. "Machine learning-based social media bot detection: a comprehensive literature review." Social Network Analysis and Mining 13.1 (2023): 20.
11. https://www.kaggle.com/datasets/davidmartngutirrez/twitter-bots-accounts
12. Waheed, Nazar, et al. "Security and privacy in IoT using machine learning and blockchain: Threats and countermeasures." ACM Computing Surveys (CSUR) 53.6 (2020): 1-37.