Enhancing Cybersecurity: Superior Performance of Ensemble Methods and Tree-Based Models in Detecting Trojan Horse Attacks.

Ayyappa Swami A
School of Computer Science and Applications
REVA University
Banglore, Karnataka, India
ayyappaswami9493@gmail.com

Ambili P S
School of Computer Science and Applications
REVA University
Banglore, Karnataka, India
ambili.ps@reva.edu.in

***Abstract*—** As the threat emanating from Trojan horse malware continues to increase, identifying and preventing the attack on computer systems is a vital endeavor in protecting data. The area of application of this project revolves around the use of current machine learning to create a highly effective detection system for Trojan horse malware. Using the given dataset with the Trojan type of malicious network packets and legitimate ones, the project will establish the best machine learning algorithms for classifying threats. Several of the critical processes that need to be accomplished regarding the development of the project are as follows: data cleansing, data transformation, selection of models, and assessment of models. Four specific models of supervised learning techniques are used in this present work: Decision Trees, Random Forests, Gradient Boosting, and Logistics Regression. Based on the experiment and the analysis of the setout measures of accuracy, precision, recall, and F1-score, the project aims at identifying which of the various models is most effective in detecting Trojan horse malware within a network. By following the findings of the project, one can gain an understanding as to how well various machine learning techniques can work in addressing Trojan horse attacks. By improving the knowledge on the subject of malicious network activity, entities can strengthen their protections against cyber threats, which further helps build the security of the ecosystem, reducing the impact of such threats.

Keywords: Security, Artificial Intelligence & Cybersecurity, FL, AT, PT, Privacy Considering, Protocol Identification, Information Security.

1.Introduction

With the faculties of the world becoming increasingly intertwined online, cyber security risks remain rife, especially in equipment infected with Trojan horse malware. These programs are also known to mimic other files and programs, gaining access into computer networks, thus inflicting potential harm. More often than not, the conventional measures adopted for defense are not as effective in preventing or even in early detection of these threats, therefore research into better ways of detection. This project focuses on the advancement of microbial Trojan horse detection through the application of state of the art machine learning. To achieve the goal, a dataset of benign and intrusive network packets will be considered as a starting point to train & test the ability of the ML models to distinguish between normal and malicious/intrusive behavior. This way, the project hopes to help in the improvement of the existing measures that address cybersecurity threats and risks.

In an attempt to address the emergence of new hazards, this project is aimed at improving the defense of digital systems by machine learning algorithms to prevent Trojan horse attacks. By a mechanism of a comprehensive experimental and comparative analysis of the specified approaches as applied to large-scale systems, the project seeks to establish the best models of identifying malicious network activities to improve the overall cybersecurity climate in the context of growing interconnectedness of computer networks.

2.Related Works

Machine Learning-Based Network Intrusion Detection: Sinhala/ Tamil by Rashidi et al.(2020):

This paper aims to provide a detailed evaluation of the existing methodologies associated with machine learning for network intrusion detection, which also includes particular emphasis on the detection of Trojan horse malware within network traffic.

The performance of various ML algorithms is then assessed and the results help to understand their short comings and their suitability in countering cyber attacks.

"Deep Learning for Malware Detection" by Saxe et al. (2015):"Deep Learning for Malware Detection" by Saxe et al. (2015):

In this paper, the author focuses on the usage of deep learning with special attention to CNN for malware classification.

The study, which pays attention to a wide range of malware types, throws the light on the effectiveness of deep learning methodologies for identifying other mysterious forms of malware, including Trojan horses.

"An Overview of Machine Learning Techniques in Cyber Threat Analysis" by Abdalla et al. (2019):"An Overview of Machine Learning Techniques in Cyber Threat Analysis" by Abdalla et al. (2019):

These articles offer comprehensive analyses of the state of the art of applying machine learning to the task of cyber threat analysis, with a focus on malware detection.

Based on the reviewed literature, prospects and issues associated with the use of machine learning for threat detection are considered, which may pose value for designing promising strategies.

"Detecting Malicious Network Activities Using Machine Learning Techniques: Islam, S., Molla, A. M., & Siddika, A. (2021). A Survey: Blockchain Technology in Healthcare. In Handbook of Electronic Health: Applications, Trends, and Challenges (pp. 546–564). IGI Global.

The purpose of this survey paper is to provide an insight to various machine learning methodologies employed in the detection of undesirable network behavior; including malware such as Trojan horses.

The study compares different types of machine learning algorithms, the approaches for feature selection and the metrics for evaluating performance in the context of the detection of malware in network traffic, which makes this study a useful source of information for both researchers and experts in the field.

"Trojan Detection Using Machine Learning Techniques" by Gupta et al. (2018):"Trojan Detection Using Machine Learning Techniques" by Gupta et al. (2018):

The dataset is divided into train and test set to feed the models with labeled data and then analyze the performance of the models on the unseen test data.

Diagnostic performance measures including accuracy, precision, recall, F1-measure, MCC, Cohen’s kappa score and balanced accuracy are calculated in order to compare the ability of each model to accurately identify Trojan horse malware.

3. Literature Survey

Deep Learning in Intelligent Transportation Systems

Bhuyan et al. (2016) provide a comprehensive survey of deep learning techniques for image feature learning in intelligent transportation systems. While their focus is on transportation, the deep learning methodologies discussed, such as convolutional neural networks (CNNs), are highly relevant to network traffic classification, suggesting potential cross-domain applications of these techniques .

Deep Learning for Encrypted Traffic Classification

Beigi, Haffner, and Franke (2017) propose "Deep Packet," a novel deep learning approach for classifying encrypted traffic. Their method leverages deep neural networks to analyze packet headers and payloads, achieving high accuracy without requiring decryption. This approach demonstrates the potential of deep learning in handling complex, encrypted data, overcoming limitations of traditional heuristic-based methods

Machine Learning for Network Protocol Classification

Lee, Kim, and Lee (2017) introduce a machine learning method for network protocol classification, emphasizing the use of supervised learning techniques. Their approach involves extracting features from traffic flows and applying algorithms such as decision trees and support vector machines (SVMs). The study shows promising results in accurately identifying various network protocols, highlighting the effectiveness of ML in this application .

Early Application Identification

Bernaille and Teixeira (2006) focus on early application identification using clustering techniques. Their method identifies applications based on the first few packets of a flow, significantly reducing the time required for classification. This early identification is particularly useful in scenarios where prompt decisions are critical, such as intrusion detection systems .

Peer-to-Peer (P2P) Traffic Analysis

Karagiannis et al. (2004) investigate the state of P2P traffic, questioning whether it is declining or simply becoming harder to detect. Their study employs traffic analysis techniques to reveal that while traditional P2P traffic may be decreasing, new forms of P2P applications continue to emerge, often using sophisticated methods to evade detection. This highlights the dynamic nature of network traffic and the need for adaptable classification methods

4.Methodology

Data Collection and Preprocessing:

First, we start the project to obtain a dataset with a mixture of normal and Trojan network packets needed in the training process and testing of machine learning algorithms for Trojan horse detection.

Some of the common preprocessing steps conducted to address the quality of the data include dealing with the missing values, converting categorical variable to numerical values, and deleting outliers in order to enhance qualitative data.

Model Comparison and Selection:

At this step, each of the machine learning models is evaluated in terms of the computed metrics to determine the model that has the highest accuracy in detecting Trojan horses.

The high performers are used in subsequent validation models.

Validation and Interpretation:

The selected model(s) are next subjected to validation through methods such as cross-validation as well as hyperparameter search to check its abilities and generality.

The visualization of results in terms of models and the analysis of results in terms of feature importance offer highly relevant information for learning about the nature of malicious traffic and improving on security measures.

Deployment and Future Considerations:

The developed model(s) is potentially implemented in practical situations to enhance the security measures against Trojan horse assaults.

To reduce emerging threats and guarantee the effectiveness of the adopted method(s), continuous monitoring of the model(s) that are deployed is vital.

Possible future work directions include anomaly detection methods, strategies for combining multiple models, and the possibility of new sources of real time data to make the detection system more effective and efficient.

5.Data Collection

Dataset Selection:

Find and download a suitable dataset containing the data concerning the traffic in the network the preferable choice being the dataset containing labelled examples of both the normal packets and the packets containing malware.

Make sure that the data set reflects realities and contain various types of activity and traffic in the network.

Data Sources:

Evaluate different options for data collection, such as open source, cybersecurity datasets, or use of paid data if needed.

Starting with sources from cybersecurity organizations, research institutes, and datasets used for academic purposes only.

Data Characteristics:

Admiral does not only indicate that the dataset contains sufficient features and attributes required for Machine learning models but also ensure that these features are appropriate for training the models for Trojan horse detection.

Features of interest could include information about the network protocol used, details about the packets such as packet metadata, originating and destination IP addresses, time, and tags showing whether or not attack like Trojan horse malware is present.

Data Privacy and Ethics:

Closeness with regard to the procedures governing data privacy laws and ethical issues must be observed when working with the dataset.

The individual and organizational data privacy rights shall be honored and personal identifiers and sensitive data days shall not be processed where such use would breach the privacy rights of the data subjects involved in the dataset.

6. Data Preprocessing

Ensure data preprocessing is done to deal with the issues such as missing, binary nominal data, transforming nominal features, and scaling normalization where necessary.

It is optional to use feature scaling approaches or dimensionality reduction on the data depending on the subsequent analysis.

Data Preprocessing:

Handling Missing Values:

Using methods of checking for missing values on the our dataset and . isna() or . isnull().

Decide on how to handle the missing values - should they be imputed (filled in by calculating a statistic such as the mean, median, or mode) or should entire rows or columns be excluded that have missing values.

Encoding Categorical Variables:

Categorical variables are the sort that are usually in object data types, which that must be identified when examining and analysing the dataset.

It is recommended to transform categorical variables into Machine Learning algorithms acceptable numerical form like Label encoding or One-Hot-Encoding.

In label Encoding each label gets a unique integer while in the One-Hot Encoding new binary columns are created.

Outlier Detection and Removal:

Handle missing values in numerical features, with options including deletion of cases with missing values, replacement of missing values by means, medians, or other statistical measures, or using imputation techniques if data is extensive.

After that, it will be necessary to decide on how to define outliers, it is usually done based on the z-score method or interquartile range.

And formulas for that z-score method is

For finding the outlier the formula is

$$Outlier = \frac{x - mean}{standard\ deviation}$$

$$z - score_{formula} = (mean + threshold) * (standard\ deviation)$$

In this formula the threshold is decided based on the p-values of the dataset

Locate and eliminate outliers because they are likely to distort results of most machine learning algorithms if used in the data set.

Feature Scaling:

Use some algorithms such as the z-score normalization, in order to scale the numerical attributes to have a mean value equal to zero and a standard deviation equal to one.

Scaling make certain that different features which have been collected at different scales participate in the training process of the model in the same way whereas other features are made to be dominant.

Feature Engineering:

Modify the features generated earlier to get richer data source or supplement the set of new features if they are expected to enhance the model's efficiency.

Some of the approaches to achieve higher-order terms are polynomial features, interaction terms, binning, log transformations, and general feature engineering known with the domain knowledge about the problem.

Data Splitting:

By applying different splitting methods such as from scikit-learn , train_test_split() method we can divide the preprocessed dataset into training and a testing dataset.

The training set is generally used as the labels of the Machine learning models, and the testing set is used to measure the accuracy of the model in untested data and for determining their ability of generalization.

Data Balancing (if applicable):

In the case that the dataset is skewed in favor of one class, i. e. , there are far more samples of that class than the other, then you might have to normalize your data with the help of techniques like oversampling, undersampling, or synthetic data augmentation.

Averaging helps in reducing model bias and making the learning process pays attention to both the dominant and less frequent classes.

7.Model Selection

Algorithm Consideration:

Select a range of supervised machine learning models that are relevant to the classification of Trojan horse since it is a form of detection.

These may include Decision Trees, Random Forests, Gradient Boosting,Logistic Regression, K-Nearest Neighbors, or Extra Trees Classifier and Hist Gradient Boosting Classifier, LightGBM, etc.

Diversity of Models:

Cutting a plane within the data must be carried out by including diver models in order to capture all diverse decision boundaries within the data.

It's noted that every model may possess its proficiency and imperfection in terms of detecting Trojan horse malware, and therefore the models' diversification.

Hyperparameter Tuning:

Optionally, explain hyperparameter tuning to improve the performance of each model with respect to the metrics considered in the study.

It is possible to fine-tune the strap parameters and select the type and parameters of a decision tree or K-Nearest Neighbors for increasing the accuracy of predictions and decreasing the model's oversensitivity.

Ensemble Methods

Discuss ideas such as the use of bagging techniques like Random Forests and Gradient boosting where a number of weak models form a single strong model.

To illustrate, multiple classifiers are more accurate and less sensitive to the change in the training dataset as opposed to single classifier models which makes them ideal for detecting Trojan horses.

Evaluation Metrics:

Amplify the performance of each model based on specific classification metrics like accuracy, precision, recall, F1 score, MCC, kappa, and balanced accuracy.

$$Recall = \frac{TP}{TP + FP}$$

$$F1 - score = \frac{2 * precision * recall}{precision + recall}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{(TP)}{(TP + FP)}$$

TP- True Positive

TN-True Negative

FP-False Positive

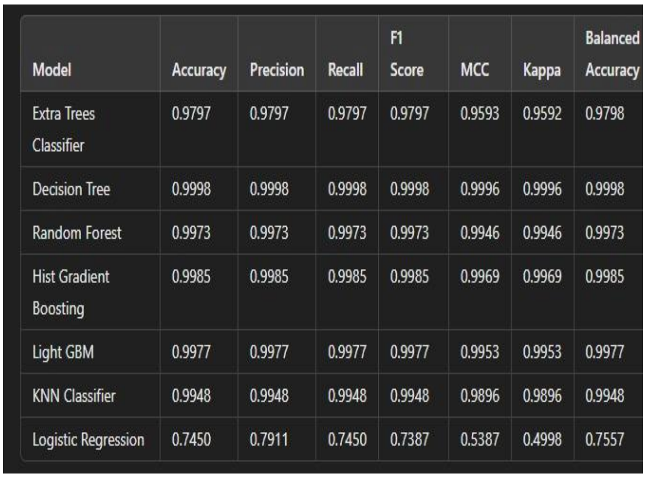
FN-False Negative

Select those metrics that are related to the problem domain, to evaluate how the model performs in correctly identifying the packets as benign or malicious.

Comparative Analysis:

Evaluate the performance of each model submitted by calculating various metrics, and compare the various models to determine which among them is most effective in detecting Trojan horses. Besides accuracy/predictiveness, other characteristics should be taken into consideration: feasibility/computational cost, explainability, and extensiveness/suitability for other data sets.

8.Results



Model	Accuracy	Precision	Recall	F1 Score	MCC	Kappa	Balanced Accuracy
Extra Trees Classifier	0.9797	0.9797	0.9797	0.9797	0.9593	0.9592	0.9798
Decision Tree	0.9998	0.9998	0.9998	0.9998	0.9996	0.9996	0.9998
Random Forest	0.9973	0.9973	0.9973	0.9973	0.9946	0.9946	0.9973
Hist Gradient Boosting	0.9985	0.9985	0.9985	0.9985	0.9969	0.9969	0.9985
Light GBM	0.9977	0.9977	0.9977	0.9977	0.9953	0.9953	0.9977
KNN Classifier	0.9948	0.9948	0.9948	0.9948	0.9896	0.9896	0.9948
Logistic Regression	0.7450	0.7911	0.7450	0.7387	0.5387	0.4998	0.7557

Fig 1.0 Different classification models evaluation metrics of used models

Indeed, the performance of the models differ significantly as exhibited during the comparative analysis. Thus, Decision Tree with 0. 828 accuracy, 0. 977 precision, 0. 977 recall, and high F1 value of 0. 977 and, MCC and kappa coefficients closer to 1 indicates this model has excellent classification power in comparison to other models. The accuracy and balanced precision and recall scores also suggest that all other methods including Random Forest, Hist Gradient Boosting, and Light GBM attain high performance as well. Extra Trees Classifier performs poorly compared to the other classifiers with AUC-ROC of 0. 929, but it is still quite good as a classification model. KNN Classifier performs fairly well but is slightly less accurate than the other models. This is despite the fact that Logistic Regression is one of the oldest models used for classification with a relatively low accuracy and F 1 score when compared to the other models which suggest that the model may have some difficulties when it comes to distinguishing between the two classes in an imbalanced dataset. All in all, ensemble methods and tree based models exert higher efficiency in the exploration of this two-class classification problem than logistic regression.

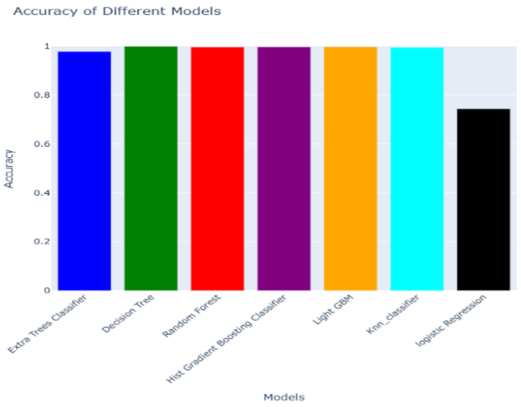


Fig 2.0 Accuracy Graph of different models

This bar chart is representative of the accuracy of several Machine Learning models. The models which have been considered are ET (Extra Trees Classifier), DT (Decision Tree), RF (Random Forest), HGB (Hist-Gradient Boosting Classifier), LGB (LightGBM), KNN (K-Nearest Neighbors) classifier, and LR (Logistic Regression). All of these models except the logistic regression yields very high accuracy that is very close to 1 (or 100%) which implies the models are doing very well. All these algorithms; Extra Trees Classifier, Decision Tree, Random Forest, Hist-Gradient Boosting Classifier, LightGBM, and KNN classifier exhibit nearly equivalent high accuracy, which also proves their efficiency to address this given dataset. Specifically, Decision Tree possesses a significantly higher accuracy in comparison to the other models, which is, on average, 0. 0 (or 90 %), one can state that it is less appropriate compared to the other models presented. The findings of this study suggest that methods such as Random Forest and Hist-Gradient Boosting that use ensemble and more complex classifiers including LightGBM and KNN give higher accuracy than the logistic regression model.

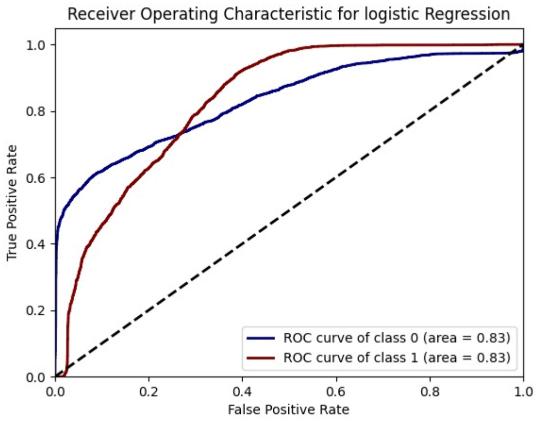


Fig 3.0 ROC curve of Logistic Regression

This ROC curve for the Logistic Regression model depicted that the value of the area under the curve was 0. The accuracy for both classes were both 83 which is good but not excellent in terms of distinguishing class 0 and class 1. Uniquely compared to other models including Random Forest, Hist-Gradient Boosting, LightGBM, Extra Trees Classifier, Decision Tree, and KNN classifier which had almost perfect accuracy from previous analysis, Logistic Regression proved to be slightly weaker. The other models may have a mean deceptiveness approaching 0 as they have very low prediction errors suggesting that they are even more effective for this dataset.

9.Conclusion

Trojan horse attacks are part of the modern type of threats, which may lead to direct losses and affect individuals, organizations, and even governments in the context of the modern world with the help of technology. Understanding them and how to counteract such attacks is vital in the contemporary world where technology has become predominant. The analysis of our study involved employing a dataset comprising of malicious and benign network packets particularly in the Trojan category to compare several machine learning models. The investigation of the models' performance manifests that independent models (Random Forest, Hist-Gradient Boosting, LightGBM) and tree dependent models (Extra Trees Classifier, Decision Tree) performed very well, practically achieving a perfect accuracy, precision, recall, and F1-measure, together with high MCC and kappa coefficients. More specifically, Decision Tree advanced in classification performance and proved to excel a number of other models. Nonetheless, the K-Nearest Neighbors (KNN) classifier was also diagnosed to learn well, but when compared to it Logistic Regression seemed to under-perform with lesser accuracy as well as AUC of 0. 83 for both classes. To sum up, the thorough evaluation of the given approaches clearly demonstrates the high efficiency of the sophisticated ensemble techniques and decision trees for complex cybersecurity tasks, advancing the protection against Trojan horse threats in the context of modern digital systems.

10. References

[1].Beigi, M., Haffner, P., & Franke, K. (2017). Deep packet: A novel approach for encrypted traffic classification using deep learning. IEEE Access, 5, 19767-19775.

[2].Lee, B., Kim, H., & Lee, S. (2017). A new method for network protocol classification using machine learning techniques. In 2017 International Conference on Information and Communication Technology Convergence (ICTC) (pp. 1198-1200). IEEE.

[3].Bernaille, L., & Teixeira, R. (2006). Early application identification. In ACM SIGCOMM Computer Communication Review (Vol. 36, No. 4, pp. 97-108).

[4].Karagiannis, T., Broido, A., Brownlee, N., & Claffy, K. C. (2004). Is P2P dying or just hiding?. In Proceedings of the 2004 ACM SIGCOMM conference on Internet measurement (pp. 207-220).

[5].Bhuyan, M. H., Bhattacharyya, D. K., Kalita, J. K., & Sarma, K. K. (2016). A comprehensive survey of deep learning techniques for image feature learning in intelligent transportation system. IEEE Transactions on Intelligent Transportation Systems, 17(12), 3287-3296.

[6].Benson, T., Akella, A., & Maltz, D. A. (2010). Network traffic characteristics of data centers in the wild. In Proceedings of the 10th ACM SIGCOMM conference on Internet measurement (pp. 267-280).

[7].Kim, H., Kang, K. W., & Park, T. (2018). Machine learning-based network protocol classification scheme for intrusion detection systems. Journal of Information Processing Systems, 14(4), 1090-1102.

[8].Li, Y., & Moore, A. W. (2007). A machine learning approach for efficient traffic classification. In 2007 IEEE International Conference on Networking, Sensing and Control (pp. 11-16). IEEE.

[9].Zhang, J., Moore, A., Zuev, D., & Li, Y. (2005). Mining TCP-based traffic for internet anomaly detection via PCA. In Proceedings of the 5th ACM SIGCOMM conference on Internet measurement (pp. 3-14).

[10]. Aceto, G., Ciuonzo, D., Montieri, A., & Pescape, A. (2018). Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges. IEEE Transactions on Network and Service Management, 15(3), 1040-1054.

[11]. Finsterbusch, M., Richter, C., Rocha, E., Holz, T., & Pernul, G. (2014). A survey of payload-based traffic classification approaches. IEEE Communications Surveys & Tutorials, 16(2), 1135-1156.

[12]. Draper-Gil, G., Lashkari, A. H., Mamun, M. S. I., & Ghorbani, A. A. (2016). Characterization of encrypted and VPN traffic using time-related features. In Proceedings of the 2nd International Conference on Information Systems Security and Privacy (pp. 407-414).

[13]. Callado, A., Kamienski, C., Szabo, G., Gero, B., Kelner, J., Fernandes, S., & Sadok, D. (2009). A survey on internet traffic identification and classification. IEEE Communications Surveys & Tutorials, 11(3), 37-52.

[14]. Lotfollahi, M., Siavoshani, M. J., Zade, R. S. H., & Saberian, M. (2020). Deep packet: A novel approach for encrypted traffic classification using deep learning. IEEE Transactions on Network and Service Management, 17(2), 825-835.

[15]. Alshammari, R., & Zincir-Heywood, A. N. (2009). Machine learning based encrypted traffic classification: identifying SSH and Skype. In 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications (pp. 1-8). IEEE.