CMP417 – Engineering Resilient Systems – Machine Learning

Finlay Reid
1904629

Contents

Introduction

Background

Supervised learning is an approach to machine learning that utilises a training set to teach models to achieve the desired result. It can be trained over time, using the loss function to minimise error and achieve the best accuracy possible. This machine learning subcategory is broken down into two distinct problems classification and regression. Classification attempts to aggregate the input into the desired number of categories based on data used in the training set. Companies utilise classification models for several tasks, including filtering email into the spam or non-spam, categorising customer feedback as positive or negative, and feature recognition(Nguyen et al., 2012).

Within the classification problem, there exist numerous algorithms and computation techniques. One of these is a neural network that relies on a hidden layer of nodes. These perform checks on whether the input has a specific characteristic to determine the path through the nodes. This algorithm imitates the interconnectivity of the human brain and has different versions, including radial basis function neural network and multilayer perceptron neural network(Verypossible.com. nd).

Another one of these classification algorithms includes the random forest method. A flexible algorithm that utilises a group of uncorrelated decision trees is known as a “forest”. Decision trees can be considered layers of condition statements merged to achieve more accurate results.

The fictional scenario involved technical staff monitoring the company network for suspicious traffic, resulting in recording data that seemed suspicious. However, the technical team did not have the required knowledge to understand it fully. Therefore, the data recorded is a sample of the network packet traffic data collected by the company. A classifier will be designed and discussed based on the aforementioned data that categorises the network packet data according to their attack category. The problem of network classification is suitable for classification algorithms due to the numerous features present within network traffic data, i.e. service, as they will help determine if a packet is harmful or not.

Aim

This project aims to design a classification model that can distinguish harmful network traffic from ordinary and categorise this data depending on its attack type.

Methodology

Classifier/Algorithm Choice

Material such as the SciKit Learn Algorithm and SAS machine learning cheat sheets assisted with selecting an appropriate algorithm for the problem of network classification(Li, H et al, 2022). These cheat sheets follow a decision tree format, including statements related to the size of the sample and type of data. Following this documentation, while considering the demands of attempting to categorise network packet data according to their attack category, it was determined that the project would utilise classification algorithms. The classification subdivision of machine learning has many algorithms, each with its strengths, weaknesses and limitations. Considering the dataset, the project’s task, and previous research into network classification using machine learning, the algorithm random forest was perceived to be suitable(scikit-learn, 2022). Mainly because the algorithm is effective with datasets that are not small, such as this one, and Random Forest is suited for multiclass problems. The other algorithm selected was the neural network algorithm, as it is

efficient for high-dimensionality tasks such as the current problem of categorising network traffic based on the attack type.

Random Forest

The random forest algorithm is an ensemble classifier that utilises many decision trees. Developed by Tin Kam Ho in 1995, the algorithm employs the bagging method. This meta-algorithm is used within random forests as it increases the precision, reduces variance and assists in evading overfitting. The algorithm's core is a group of random decision trees; first new data sets are created using the original data set, known as bootstrapping. The data sets are used to train the decision trees independently. However, random features from each dataset are selected for a tree. The trees can be generated after creating the new data and selecting the subsets of features from each dataset. Random forest utilises aggregating to determine the prediction, essentially meaning random forest combines all the results from each tree and looks at what is the majority result. As previously mentioned, the algorithm utilises bagging, which combines bootstrapping and aggregation.

One advantage of the Random forest algorithm includes being one of the most accurate algorithms due to its use of ensemble learning and bagging. It also has excellent stability, with the algorithm's ability to handle new data points as the new data may impact one tree, but it is tough to impact all the trees. Furthermore, the algorithm is effective when attempting to estimate data and making accurate predictions if a large portion of data is missing. On the other hand, considering the algorithm's limitations, the random forest can be computational heavy by using numerous decision trees to determine the output from aggregation. Furthermore, the algorithm requires much time to train due to its many decision trees.

The random forest is effective for the problem of classifying network packets as it is efficient with large datasets. This is relevant because it is highly likely there would be thousands upon thousands of network packets that would make up the data, requiring the classification of this network traffic. When categorising the data, the chosen classifier must be highly accurate. If not, the appropriate mitigation techniques cannot be developed for that attack. Random forest is one of the most accurate classifiers; this would ensure traffic classification was completed with minimal errors. Compared with other algorithms when classifying network data, random forest currently is one of the fastest, as evidenced by research undertaken by the Queensland University of Technology(Perera et al.,2017).

Neural network

Neural networks are a group of algorithms based and closely tied to the human brain. The first implementation of a neural was developed in 1958 by Frank Rosenblatt, but there had been a continuous progression related to the concept of neural networks long before. Neural networks are made up of nodes, and each layer is responsible for different tasks. Layers include the input layer, hidden layer and output layer. The input layer is in charge of interfacing with the outside environment and receiving the inputs. Hidden layers lie between the input and outer layer; these layers are responsible for processing the data from the input layer, essentially taking the required features from the inputted data. Finally, the output layer deals with coalescing the data and presenting it.

An advantage of neural networks is their ability to adapt and learn, helping the algorithm produce output that differs from the original data. If a neuron that makes up a network is unresponsive, the network can diagnose this problem while still producing an output, meaning the method has fault tolerance. However, neural networks are not without their limitations, as this algorithm requires a decent level of computation power, including processors with parallel processing power. Another drawback when using neural networks is the uncertainty in the structure, as there is no specific rule for determining the structure of artificial neural networks. Finally, the network relies significantly on the data fed to it. The more considerable amount of data used during training, the more accurate the results are.

Like the random forest algorithm, the neural network is highly effective when using large datasets. As previously mentioned, thousands upon thousands of network packets would make up the data, ensuring that a weakness of neural networks is negated and the model can be trained effectively. Another reason for using neural networks for network classification is the algorithm's ability to deal with high dimensional data and problems. As network packet data would most likely have hundreds of features related to one packet, the model can use this data to determine if the packet is harmful. Finally, graphics processing units are utilised to perform similar calculations in parallel within neural networks. This technology constantly improves and evolves, ensuring these algorithms become more efficient over time. Random forest can be improved by upgrading hardware, presumably not a problem for a company looking to classify network traffic.

Classifier Design

The random forest algorithm was chosen with consideration of the advantages and limitations discussed in the previous section. When creating a classifier, several steps must be adhered to to ensure the model functions as intended; these include data ingestion/preprocessing, modelling, analysis and communication of results.

Data ingestion/Preprocessing

This step in the classifier is responsible for extracting and preparing the data ensuring that the machine may easily parse it. It is a crucial step in a machine learning model as mistakes here can hugely affect/skew the results. In the context of the network classification problem, a sample of network packet traffic has been captured, which will be used as the training dataset. The excel file provided details information related to the protocol, service, ID and more. The attack category field is present in the file and Is essential in being able to train the model. Nine network packet categories are described in the paper and are used to sort the data(Moustafa, N. and Slay, J., 2015). Data would most likely have to be converted, ensuring there was no non-numerical data.

Modelling

This step in the machine learning framework is where the model's training is completed. First, mathematical knowledge is applied to train the model based on the network packet dataset enabling the model to make predictions. It is a crucial step that will determine the quality and accuracy of future predictions. Then, using the nine classifications of network data, the algorithm can be created to analyse the network data, building separate trees to sort the different types of network traffic.

Analysis

The analysis section of the model is responsible for analysing the results, essentially putting the model into practice and allowing it to make predictions. It approximates how the model will perform in the real world. When applied to the issue of network classification, the analysis will determine if

any of the network traffic is suspicious and what attack category the attack is. This allows the staff in the technical department to see clearly what type of attacks the network has been flagged

Communication of Results

Communicating the results involves presenting and communicating the results of the predictions. This is where the results of classifying the network data are presented

Evaluation

One of the most critical tasks in creating a machine learning model involves evaluating its performance. These evaluation metrics are closely linked with machine learning and differ depending on the current task, classification or regression. Therefore, it is possible to significantly improve the model's accuracy by utilising several unique metrics for performance evaluation. Evaluation metrics include but are not limited to the confusion matrix, Receiver Operating Characteristic curve and Area Under the Curve. The confusion and the Receiver Operating Characteristic curve evaluation process will be discussed in this section.

Confusion Matrix

The confusion matrix is an evaluation method compromising a table that presents the correct and incorrect predictions calculated by the model. This is contrasted with the classifications in the test set. The confusion matrix essentially tests the model's performance, ensuring the model can be improved. Four outcomes can be determined while performing classification predictions:

- True Positives: Number of outcomes that are positive and are predicted to be positive.
- True Negatives: Number of outcomes that are negative and are predicted to be negative.
- False Positives: Number of negative outcomes but predicted positive. Also called Type 1 Errors.
- False Negatives: Number of positive outcomes but predicted negative. Also called Type 2 Errors.

Four metrics can be discovered through this classification method: accuracy, precision, recall, and F-measure. For example, the accuracy can be calculated using the equation seen below in **Figure 1**.

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

Figure 1 - Accuracy Equation

The precision of the model is calculated using the ratio of true positives against all the positives that the model predicted. The equation used to calculate this is shown below in **Figure 2**.

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

Figure 2 - Precision Equation

Recall, also known as sensitivity, is the ratio of true positives against the positives in the dataset. It can describe the model's capacity to discover positive samples. The equation to calculate this is shown below in **Figure 3**.

$$TPR = Recall = \frac{TP}{TP + FN}.$$

Figure 3 - Recall Equation

ROC

The ROC curve is a method of evaluation involving a graphical plot used to demonstrate the relationship between the actual positive and false positive rates in an easy-to-understand visual format. When looking at a ROC graph, a higher X-axis value stipulates a greater number of false positives than true negatives. At the same time, a greater Y-axis value shows a greater number of true positives than false negatives. The decision threshold(the threshold for deciding whether a prediction is labelled “true” or “false”)depends on the ability to balance between false positives and false negatives. AUC(area under the curve) can be used to measure the accuracy of the classifier and can be used to compare different models since it summarizes the data from the whole ROC curve. An example of a ROC graph is presented below in **Figure 4.**

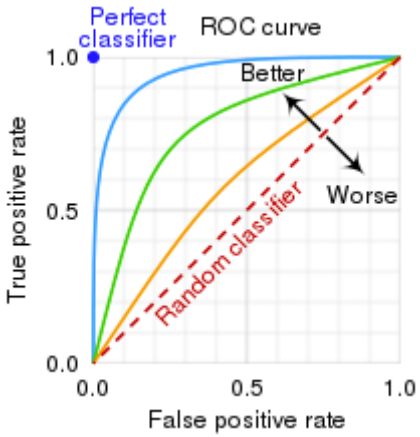

Figure 4 - ROC Graph

References

Baheti, P., 2022. *A Simple Guide to Data Preprocessing in Machine Learning*. [online] V7labs.com. Available at: <https://www.v7labs.com/blog/data-preprocessing-guide#what-is-data-preprocessing> [Accessed 26 April 2022].

Bhatia, R., 2022. *When not to use Neural Networks*. [online] Medium. Available at: <https://medium.datadriveninvestor.com/when-not-to-use-neural-networks-89fb50622429> [Accessed 26 April 2022].

Education, I., 2022. *What is Supervised Learning?*. [online] Ibm.com. Available at: <https://www.ibm.com/cloud/learn/supervised-learning> [Accessed 26 April 2022].

Mayo, M., 2022. *Frameworks for Approaching the Machine Learning Process - KDnuggets*. [online] KDnuggets. Available at: <https://www.kdnuggets.com/2018/05/general-approaches-machine-learning-process.html> [Accessed 26 April 2022].

Moustafa, N. and Slay, J., 2015, November. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In *2015 military communications and information systems conference (MilCIS)* (pp. 1-6). IEEE.

Nguyen, T.T., Armitage, G., Branch, P. and Zander, S., 2012. Timely and continuous machine-learning-based classification for interactive IP traffic. *IEEE/ACM Transactions On Networking*, *20*(6), pp.1880-1894.

Li, H., Li, H., Chase, C., Paula, L. and Sglavo, U., 2022. Which machine learning algorithm should I use?. [online] The SAS Data Science Blog. Available at: <https://blogs.sas.com/content/subconsciousmusings/2020/12/09/machine-learning-algorithm-use/> [Accessed 26 April 2022].

Perera, P., Tian, Y.C., Fidge, C. and Kelly, W., 2017, November. A comparison of supervised machine learning algorithms for classification of communications network traffic. In *International Conference on Neural Information Processing* (pp. 445-454). Springer, Cham.

scikit-learn. 2022. *Choosing the right estimator*. [online] Available at: <https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html> [Accessed 26 April 2022].

Verypossible.com. nd. *Machine Learning vs Neural Networks: Why It's Not One or the Other*. [online] Available at: <https://www.verypossible.com/insights/machine-learning-vs.-neural-networks> [Accessed 26 April 2022].