**Project Report**

**Network Intrusion detection system in wireless body area network with decision trees**

**Introduction ->**

We have made a Network intrusion detection system with decision tree.

**Dataset Used ->**

We have used KDD99 dataset which includes all types of different varieties of attacks including DOS (denial of service), R2L(remote 2 user), Probe(port scanning), U2R(local 2 root).

**Extracting the features ->**

At first we filter the data, checking which features are having high correlation or having more than 25% of it's data is missing, so we drop those columns and then we divide the data into 2 parts, numerical data and categorical data. Numerical data can be compared directly because it's of the format of integer or float but categorical data is of object format so we uses dummy variable to classify all it's values in binary format. We are having 43 features but after including categorical data also we have 143 features so we used selectKBEST for feature selection and we extracted 15 best features which really affects the dataset and we remove the rest features columns.

Now we have filtered our dataset which means it is ready to used upon some model.

**Training the data with already present libraries ->**

Now we used DecisionTreeClassifier library already present in sklearn to train our data and to check what is ideal accuracy which comes and after training, we predicted that we get 87% accuracy with the data.

**Decision tree from scratch with tweaking parameters ->**

So now coming to the algorithm which we have used, Decision tree.

We have implemented it from scratch by varying the parameters to achieve more accuracy.

We have used Giny impurity to find the impurity present in a set.

Gini impurity -> probability of incorrectly classifying a randomly chosen element in the dataset. It is used to indicate the impurity/randomness present in a set.

Information Gain -> Information Gain depicts the amount of information that is gained by an attribute. It tells us how important the attribute is. Since Decision Tree construction is all about finding the right split node that assures high accuracy, Information Gain is all about finding the best nodes that return the highest information gain.

So we have used gini impurity to find the impurity present in the root node and then we find average weightage gini impurity of it's child nodes and then subtracting the 2nd from the 1st will give us the information gain.

For eg. - root node has 64% impurity present and child nodes have 50% average impurity so we got 15% information gain.

Each node takes a list of rows as input. To generate a list fo questions, we will iterate over every value for every feature.Each of the these becomes a candidate for threshold we can use to partition the data.So at first we find the best question to split our dataset into 2 parts having maximum information gain and the question will divide the dataset into 2 parts stating whether it is true or false.

After finding the best question to split the dataset, we have splited the data into 2 parts and then using the recursion on it's child nodes until we come across a base condition in which there is no information gain or information gain = 0 so then we stop at that point and then returns a dictionary showing what the output can be.

**Result ->**

And since our tree is ready, we can use it to ask any question on it now and check what is the output and compare it to actual result so this way we got an accuracy of 89% which is quite good to ideal accuracy of 87%.

Since our model is ready, so we used pickle to store the model so we dont have to train the model again and again.

After that we have calculated perfomance metrices to show the results.

**Made by**

**Manmohan Prajapat 106119072**

**Kartikey Agarwal 106119058**