Preventing Black Hole

Attacks In Manets Using Dynamically Generated Audit Data

|  |  |  |
| --- | --- | --- |
| Authors | Version | Date |
| Thi Doan, Alan Gaugler, Tristan Hore, Johnsons Sangah | 1.0 | April 2023 |

     Approval by Team Members:

|  |  |
| --- | --- |
| Team Member Name | Student ID |
| Thi Doan | u3090674 |
| Alan Gaugler | u885853 |
| Tristan Hore | u3218682 |
| Johnson Sangah | u3082492 |

Contents

[Executive Summary 3](#_Toc132979231)

[Introduction and Description of the Project 3](#_Toc132979232)

[Assessment of the project deliverables 4](#_Toc132979233)

[Project Proposal and Plan 6](#_Toc132979234)

[Source code for the NS-3 simulations 6](#_Toc132979235)

[Data traces produced from the NS-3 simulations 7](#_Toc132979236)

[Datasets Produced from the Data Traces 7](#_Toc132979237)

[Routing Table which Identifies Malicious Nodes Not completed 14](#_Toc132979238)

[Report on resources. 14](#_Toc132979239)

[Outstanding Issues 14](#_Toc132979240)

[Report of risks mitigated. 15](#_Toc132979241)

[Report on lessons learnt (one issue must be an ethical issue). 15](#_Toc132979242)

[Handover Materials to the Sponsor 16](#_Toc132979243)

[Recommendations to the Sponsor 17](#_Toc132979244)

[Opportunities for Future Development 18](#_Toc132979245)

[Sponsor Signoff 19](#_Toc132979246)

# Executive Summary

This project was initially intended to have a duration of one year as the initial scope was very large. All team members were only able to accommodate the project duration of one semester. On our first meeting with the project sponsor and mentor we agreed to terms of reducing the scope of the project to that what is mentioned in the Project Proposal and Plan. At the end of the project timeframe, we were successful in most points of the scope.

Setbacks due to the complexities of setting up and accurately simulating the AODV protocol in MANETs and the black hole nodes in the NS-3 simulator meant that not all of the objectives could be successfully completed within the given timeframe of one semester. However, the NS-3 simulation that were produced, although not completely accurate did prove to be very useful and could be used to continue to the next stage of the process.

The produced network simulation files could be used to develop a complex script that would read the AODV protocol messages between neighbouring nodes from which training and test datasets could be created. These datasets consist of several key features that can be used to train machine learning algorithms to detect if the neighbour node is a malicious black hole node.

Some AODV protocol errors in the simulations meant that the data was not behaving completely as expected and there were difficulties in simulating black hole nodes. The values of several features belonging to black hole nodes were altered to what would be a realistic expectation if the black hole simulator had been implemented correctly.

A machine learning script was developed to read the datasets. Exploratory data analysis was conducted on these datasets to determine a final feature set that would be used for the machine learning process. Two models were developed. A random forest classifier and a support vector machine classifier. Both models proved to be very accurate on the given datasets, detecting with 100% accuracy the black hole nodes.

Naturally, much more further testing is required once the NS-3 simulations are accurately simulating the AODV networks and black hole nodes. The next steps are summarised in the **Opportunities for Future Development** section. However, if the black hole nodes simulations behave as is expected, it would appear that our developed machine learning solution can successfully be applied to the network data to determine if there are malicious nodes present in the network. With further development, this solution could be implemented into AODV network nodes.

# Introduction and Description of the Project

Mobile Ad-hoc Networks (MANETs) are decentralised networks consisting of low-power mobile devices (nodes) that can communicate with each other without the need for centralised infrastructure. This makes these networks useful in disaster situations where infrastructure is disrupted or in dynamic environments such as communication between autonomous vehicles. Ad hoc On-Demand Distance Vector (AODV) is a reactive routing protocol commonly used MANETs. However, these networks are inherently vulnerable to various security threats, including Blackhole attacks. In a Blackhole attack, malicious nodes falsely advertise that they have the shortest path to the destination node, causing all traffic to be diverted to the malicious node, which then drops or modifies the packets.

The structure of AODV protocol MANETs presents several technical challenges in securing them from Blackhole attacks. First, MANETs lack centralised authority, making detecting and preventing Blackhole attacks difficult. Second, the dynamic topology of MANETs makes it difficult to maintain trust relationships between nodes, and previously honest nodes can become malicious. Third, nodes in MANETs have limited resources, and any security mechanism implemented should not consume excessive resources. Fourth, MANETs do not have a fixed infrastructure, and nodes rely on each other to relay messages, meaning that any security mechanism implemented should not rely on a fixed infrastructure. Finally, the route discovery process in AODV makes it easy for a malicious node to launch a Blackhole attack by falsely advertising that it has the shortest path to the destination.

Trust-based mechanisms, secure routing protocols, intrusion detection systems, and game-theory-based approaches have been proposed to address the issue of Blackhole attacks in MANETs. This project used machine learning (ML) approaches to improve the reliability of the MANET under attack by detecting and isolating the malicious nodes responsible. We simulated simple MANETs in the NS-3 network simulator and then captured the network traffic from these simulations. The captured data was used to produce datasets for training machine learning models to detect Blackhole attacks and identify malicious nodes.

# Assessment of the project deliverables

The initial project deliverables were as follows, we give further details of the deliverables and the progress on each in the following sections.

1. To carry out a literature review focusing on mobile ad-hoc networks (MANETs) and their vulnerability to blackhole attacks, to find ways to identify and mitigate them.  - Insert Link
2. To set up MANET, using network simulator 3 (NS-3) simulations of a black hole attack from malicious nodes to identify and detect which nodes are malicious.
3. To generate a dynamic audit data table fed by the algorithm that will try and identify the malicious nodes and measure the effectiveness of the applied algorithm in preventing network performance degradation in the event of a black hole attack.
4. To carry out a data analysis of the trace from the simulations to find opportunities to mitigate blackhole attacks and contribute to the growing academic literature in this field.
5. Pre-process and prepare the collected simulation data for use in machine learning modelling.
6. Develop and evaluate reusable machine learning algorithms that identify, detect and respond to malicious nodes in the network from the trace data provided.
7. Produce a final project report and poster on our observations and findings to help contribute to the existing academic literature.

**Literature re**vi**ew**

An extensive literature review of MANET’s (Mobile ad hoc networks), AODV (Ad Hoc On-Demand Distance Vector) protocol and typical black hole cyber-attacks was carried out by the team. This in-depth review was essential in correctly understanding how nodes in MANET’s, in particular those that employ the AODV protocol behave and establish communication. It was also essential to understand how a black hole node would infiltrate the network and interact with other nodes, acting as a transfer point between nodes but dropping the data packages, causing disruption in essential communications.

Many videos were also observed by the team which proved to be essential in learning of the protocols and the configuration of the NS-3 network.

From the literature review, a solid understanding of MANETs, the AODV network protocol was learned as well as the behavioural characteristics of the of black hole nodes infiltrating the network. Several of these characteristics could be used for a node to identify a 1st tier neighbour as a black hole node rather than a normal node.

The identification of these main unique behavioural characteristics of black hole nodes led to the development of a dataset that could be used for the machine learning process for malicious node detection.

Please refer to the literature review in the project deliverables for a more extensive description and for a bibliography of the most useful research papers reviewed for this project.

## Project Proposal and Plan

The project proposal and plan was completed and delivered before its deadline on Friday 10/03/23.

## Source code for the NS-3 simulations

Three scripts were used in the preparation of the NS3 simulation. In addition, a simple blackhole model was used early on to gain an understanding of what the larger situation behaviour should be.  
The three scripts for the man simulation were modifications of manet-routing-compare.cc, AODV-Routing-Protocol.cc and AODV-Routing-Protocol.h.

The original MANET Compare code is a script to compare the performance of different MANET routing protocols under the same conditions. This script was written in 2011 by Justin Rohrer at the University of Kansas [REF] and released under a GNU General Public Licence. This part of the project aimed to compare different implementations of AODV with and without blackhole attacks using a modifed version of this script. The modifications create the code to allow for a Blackhole attack simulation by creating malicious nodes using the routing protocol.

The simulation also relies on changes to the NS3 modules based on the work of Shalini Satre and Mohit P. Tahiliani. [REF] These changes were modifications of the original AODV (Ad hoc On-Demand Distance Vector) routing protocol implementation in the NS-3 network simulator. Changes were made to the standard files AODV-Routing-Protocol.cc and AODV-Routing-Protocol.h that are shipped with NS3.

The modified implementation adds a new attribute, "IsMalicious", to the AODV routing protocol class. The attribute is used to mark a node as malicious or not. Two new methods, SetMaliciousEnable() and GetMaliciousEnable(), are also added to set and retrieve the "IsMalicious" attribute value.

In the modified implementation, a few changes are made to the behaviour of the AODV routing protocol when a node is marked as malicious:

When a malicious node receives a packet, it simply drops it instead of forwarding it.

When a malicious node receives a route request (RREQ) message, it creates a false routing table entry with a higher sequence number and a lower hop count, effectively advertising itself as having the shortest path to the destination. The malicious node then sends a route reply (RREP) message to the source node, causing the source node to route its packets through the malicious node.

These modifications allow the simulation of Blackhole attacks in an NS-3 network, enabling the study of the impact of such attacks on network performance and developing countermeasures to protect the network.  
  
Changes in the NS3 simulator between the time of the writing of the manet-routing-compare.cc script and the NS3 installation used in this project caused a number of issues and required substantial debugging in addition to the changes needed to implement the blackhole simulations. Further changes were needed to match the changes needed in the AODV-Routing-Protocol.cc and AODV-Routing-Protocol.h scripts which were written for NS.3.25 with the versions that were included in the current installation used for the project. See the sections on lessons learned and on issues encountered in the project for further details on these travails.

Further detailed explanation on the function of the scripts and their use is included in the additional resources documents. [LINK]

## Data traces produced from the NS-3 simulations

The NS3 simulator provides a trace system that captures various simulation parameters and data and exports to various file formats using built ihelper functions. The initial plan was to produce out in the widely used pcap format. However, we experimented with producing both flat ascii output and using the built-in flowmon functions, which is the preferred method in the latest simulator version. We concluded that the pcap files were the easiest to work with using external tools such as Wireshark and so this was adopted as the main output format for the network traces. Unfortunately, substantial difficulties in extracting the correct details from the modified state flags set on nodes in the simulation caused significant delays, eventually frustrating the fulfilment of the project goals.

## Datasets Produced from the Data Traces

The key behavioural characteristics of black hole nodes in AODV (Ad Hoc On-Demand Distance Vector) networks that were identified in the literature review were some of the key features to be incorporated into the dataset to be used in the machine learning process in the detection of black hole nodes.

The data is created from running simulations, so collecting a sufficient amount of data for training and testing machine learning models is not an issue.

The output of the trace files consisted of messaging files from each node. These were in pcap file format and they were loaded into Wireshark where they could be observed. Wireshark is a free and open-source network protocol analyzer.

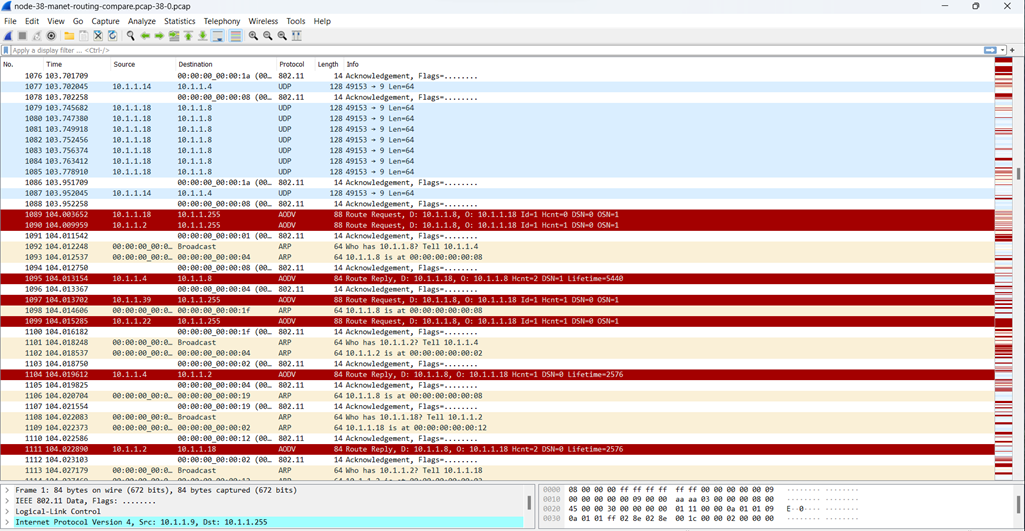


Figure 1 – Example display of AODV messaging in Wireshark

From Wireshark, these files were exported as json files where they can be loaded into a text editor or an IDE to be viewed an analyzed. The tool used in this project was Visual Studio. From here, the key message components of the AODV messages could be located.

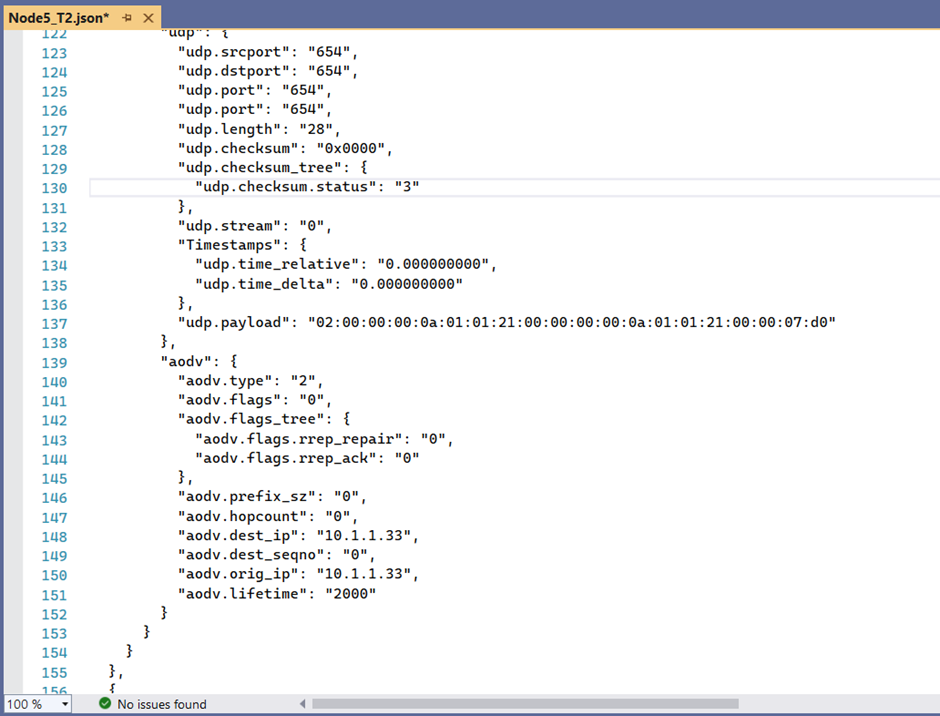


Figure 2 –Example of AODV messaging in JSON format view in Visual Studio

From this, the next stage was to extract the relevant features of the AODV messaging being sent and received from each node to and from its tier 1 neighbours and convert them into a dataset that could be used for the machine learning process.

Our team developed a Python script in Jupyter Notebooks to do this.

The list of black hole nodes was manually entered into the script. When run, the target variable “Black\_Hole\_Node” was modified to True if the neighbour node was in the list. The rows in the dataset where the black hole node was the subject node were removed from the dataset because the purpose of this process is for a normal node to learn how to detect the behaviour of a black hole node.

This process was basically split into three stages.

**Stage 1:** Read in the relevant AODV information from each node (each JSON file) and store them in separate data frames. If it is desired in the future to use another source of AODV messaging as the input into the dataset creation process, only stage 1 needs to be modified to get the relevant message features from the other source. Stages 2 and 3 can be left unchanged.

Looking at the example Stage 1 output in Figure 3, each row refers to each extracted message marked by a frame time. It contains information of the Subject Node, Neighbour Node, the AODV Message type and many other relevant features. Columns B to O are the data frames for each node produced by Stage 1.

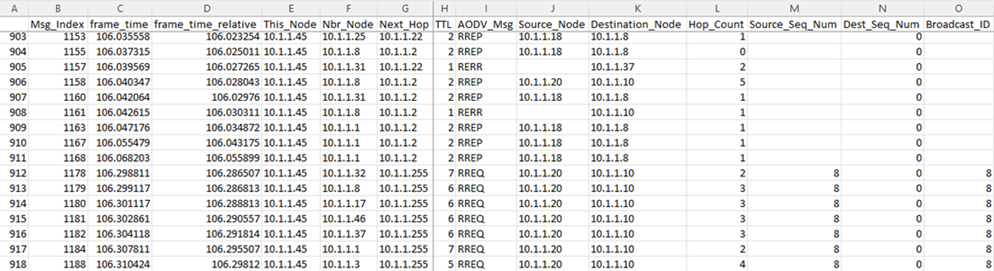


Figure 3 – Example Output of a Data Frame after Stage 1

**Stage 2:** Some processing is made for each AODV messages, creating new features in the data frame which are columns P to AA in the Stage\_2.csv files. Many of these are Boolean values of if the message is a certain type or is addressed to or originates from the neighbour node.

If the message is a RREP message responding to a RREQ message, the RREQ message index is found in Column U and the response time between the RREQ and the RREP is noted in columns V & W. If it is an RREP, the destination sequence number increment is also determined in Column Y.

As currently there are some errors with the NS-3 outputs in emulating the AODV protocol, some rows were deleted from the data frames including for now RREP-ACK messages and RREP messages from which no corresponding RREQ message could be found.

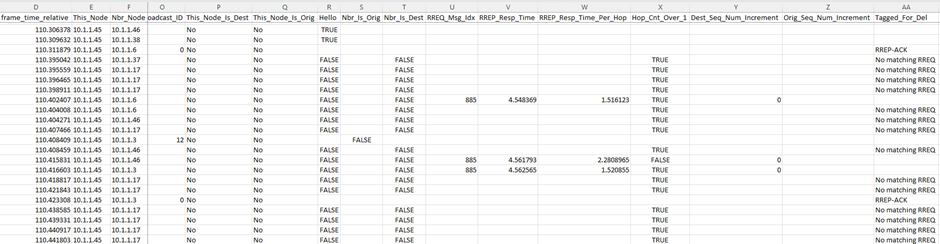


Figure 4 – Example Output of Additional Features of a Data Frame added after Stage 2

**Stage 3:** This stage will convert the data frames from Stage 2 into datasets for each Subject Node and then marge them all into one combined dataset. Every row represents features between the subject node and each of its 1st tier neighbouring nodes. Each subject to neighbour node relation is only ever one row. The features are mainly counters, percentages as well as Boolean values. All the features are described in the excel file “Dataset Features.xls” which is attached with the deliverables.

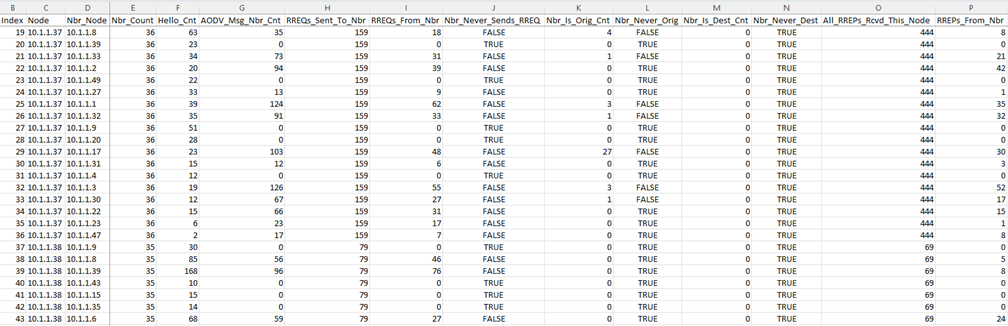


Figure 5 – Example Output of the Dataset produced after Stage 3

This script is included in the project deliverables with the name of “AODV\_to\_Dataset.ipynb”.

**Python Code for the Machine Learning Process**

A machine learning script was developed to use the datasets created by the conversion script as input. The ML script would be used to develop models to detect which nodes within the AODV network were malicious black hole nodes.

An example training and test csv files have been attached in the deliverable package to demonstrate the machine learning script working. Again, it must be highlighted that many AODV characteristics have not been accurately implemented into the datasets. Black hole nodes were not implemented accurately either and so the datasets have been modified for the rows where the target variable “Black\_Hole\_Node” is set to True. Many of the final features’ values have been altered so that the black holes behaviour is considered “dumb” and its characteristics are extremely obvious. This would be the behaviour of a black hole node that is not expecting the network nodes to have any intelligence in detecting and mitigating malicious nodes as is the case for black hole nodes.

The training and test datasets were loaded into the script and some cleaning of the datasets was performed. The Boolean values are converted to integers and any rows with missing values (mainly due to neighbours not carrying any traffic) are removed.

The features with the highest correlation to the Black\_Hole\_Node target variable were selected the train and test sets that will be used in the modelling. After evaluation, a final set of 10 features was selected to be used in the modelling.

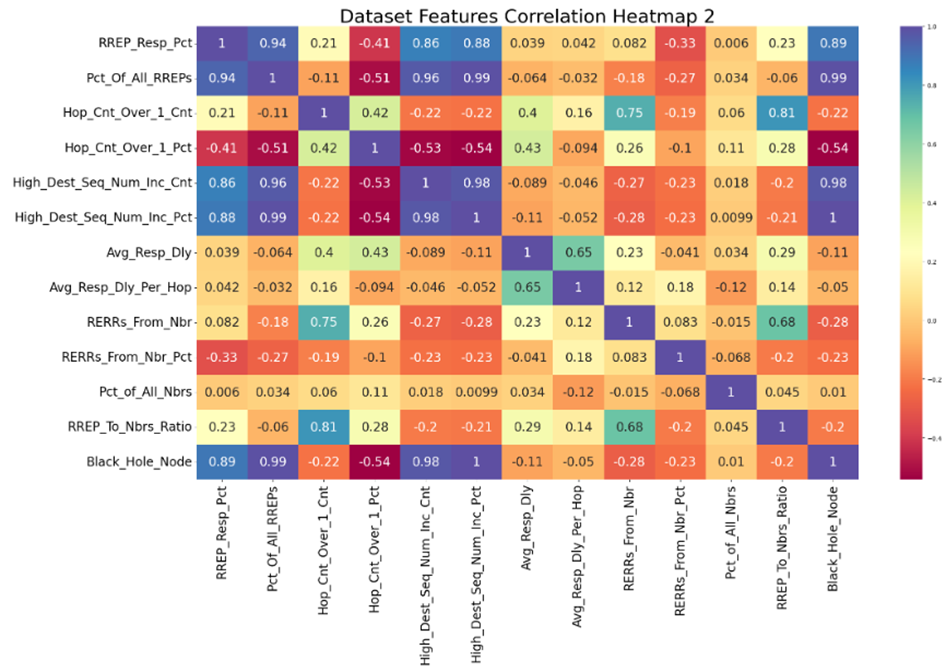


Figure 6 – Example Output of the correlation heatmap demonstrating the relationship between the features and the target variable (bottom row)

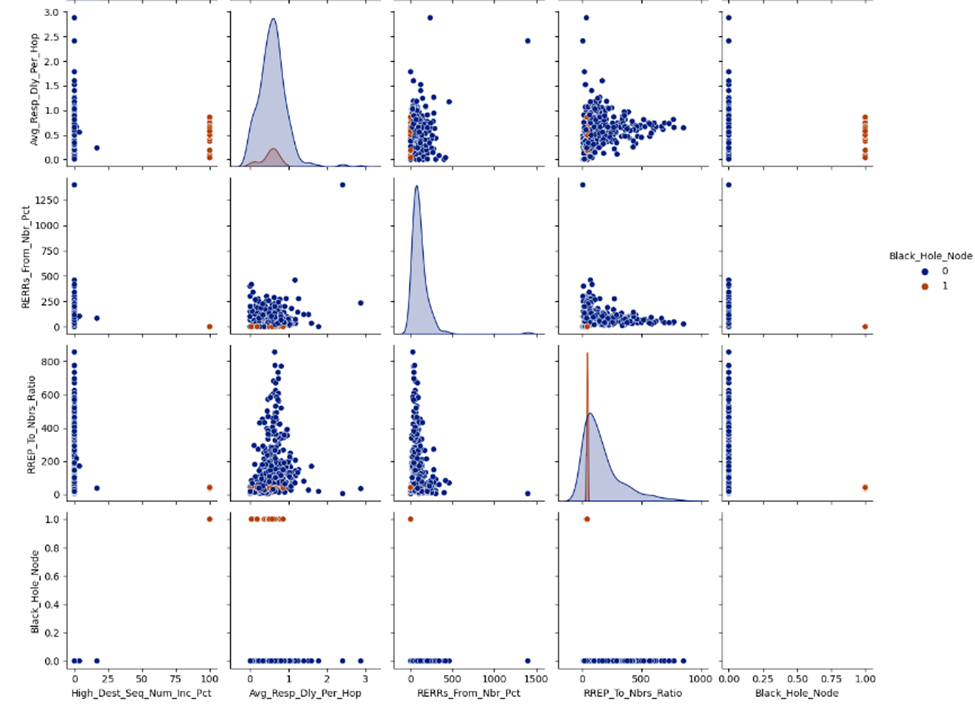


Figure 7 – Example Output of Pair Plots showing the relationships between variables.

Black hole node is brown, Normal node is Blue.

Two models were used to detect the black hole nodes. Model 1 is a random forest classifier and Model 2 is a Support Vector Machine (SVM) classifier. A grid search was carried out on both of these models to determine the optimum hyperparameter settings. Standardization was also applied to the data for the SVM so that all features carry an equal weight in the decision process. The models were trained and were then applied to the test set to determine their accuracy. In both models, their accuracy is 100%.

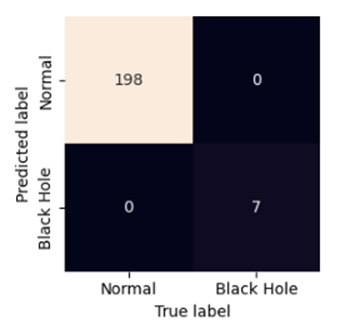


Figure 8 – Example of the Confusion Matrix from the SVM Model Classifying Black Hole Nodes.

As mentioned, much more work needs to be done once the NS-3 simulator is simulating AODV networks accurately. The Future Work section mentions what work should be carried out if anyone continues to work on this project.

This machine learning script is included in the project deliverables with the name of “ML\_for\_Malicious\_Node\_Detection.ipynb”.

**Trained Model Files**

Two models were trained to detect black hole nodes operating in an AODV network.

1. A random forest classifier
2. A support vector machine classifier

Both of these models have been ‘pickled’ as binary files and have been attached with the deliverables. They can simply be loaded using a python script and used again on network simulation datasets. This will save time having to collect and remodel the data.

## Routing Table which Identifies Malicious Nodes Not completed

The project plan was to use the ns3-AI module to deploy ML models in the simulated nodes in MANET, to mitigate the effects of a Blackhole attack. Once the AI detects the malicious nodes, the return value would be used to update the routing table to prevent further disruption by avoiding the affected nodes. The effectiveness of the ML models in preventing network performance degradation in the event of a Blackhole attack would be determined. This objective was perhaps overambitious, and given the delays caused by difficulties learning the NS3 software and debugging the simulation code it was not achievable in the given timeframe.

# **Report On Resources**

The scope of this project did not require many resources to be used. None of the allocated budget was used. The total cost of resources was $0.00. There were no hardware requirements. All the software used to develop this project was run on our personal computers.

All software used to develop the project was open source and free.

NS3 environment used.   
Host OS WSL2 on Windows 11

Linux OS Ubuntu 20.04.5

Python 3.8.10

NS3 3.37

The software used in the development of the dataset conversion script and the machine learning algorithms were:

Jupyter Notebooks utilising Python 3.10 and many add-on packages including:

Pandas

Scikit-Learn

Matplotlib

Seaborn

# Outstanding Issues

Current observed issues in the network simulation output accurately simulating AODV network and black hole node behaviour:

After running the NS-3 simulations. A detailed examination was carried out to determine if the output conforms to AODV protocol behaviour. Unfortunately, numerous issues were found, at least 13. Most of these were many issues with the messaging and protocol behaviour not strictly following AODV protocol as well as issues simulating black hole nodes. Unfortunately, the duration of the project did not permit us enough time to correct these issues.

A complete list of known issues is detailed in the Outstanding Issues section of the Project Closure Report.

Data Conversion script and Machine Learning script:

There are currently no known issues with these scripts. Once the NS-3 network simulator is producing accurate simulation trace files for each node, the logic of the data conversion script and the machine learning script must be re-verified and thoroughly tested to ensure they are correctly processing the pcap files. Undoubtedly there will be several improvements or features that can be discovered and implemented in these scripts as they are used and tested more.

# Report of Risks Mitigated.

Encountered Risks

**Technical Difficulties -** Likelihood: Moderate. Impact: Severe

Several difficulties with configuring the NS-3 environment and building and running the scripts.

Mitigation - Two different environments were setup on two different machines.

**Personnel Issues -** Likelihood: Moderate. Impact: Moderate

A team member became sick for several weeks, affecting his ability to work.

Mitigation - Overlapping responsibilities with other members. Two members were simultaneously working on the NS-3 simulations.

# Report on Lessons Learnt

**The importance of a good risk management plan:** Having a good risk management plan is crucial for projects such as this. With a relatively short timeframe for this project and a lot of challenging technical tasks it is very likely that one serious technical hurdle could cause lengthy delays and so ensure that the not all of the project’s scope can be completed. This was the case with difficulties encountered with running the NS-3 simulations. We mitigated this to a certain extent by having two team members work on these simulations. One member was successful in generating simulation files which could be edited and then used to complete the data conversion and the machine learning processes. Achieving these objectives greatly helped to make this project a success.

**Ethical Issues:** All software and scripts used were open source and all the data used in this project was generated from scripts and it nobody’s real or personal data. As such, there are no ethical issues with the actual source or privacy of the data.

One ethical lesson learned with the use of the data is to always be honest and upfront with its integrity. As has been mentioned due to simulation issues, in the datasets we modified the values of several features for black hole node rows to accurately represent the behaviour of a black hole node. This has been done as accurately as possible and we believe the values are very realistic. However, it is always imperative to be ethical and honest about any changes we have made. We have clarified this to our sponsor and mentor, stating that this was the best option for us to proceed with the project. Of course, once the NS-3 simulations are corrected, the unedited simulation data would be used, and the models’ training and testing would be reassessed.

**Practical limitations on sharing work**

Teams and shared documents are very useful for report writing, however joint programming is more difficult to share effectively remotely. The lack of access to a shared programming environment caused difficulties in sharing ideas about how to deal with difficulties working with the NS3 simulations, and limited other team members insight into the state of the simulation code.

**Trying to parallelise the work was a good decision**

The simulation task was structured in such a way that initial data for a vanilla network was produced early on. This enabled the work on data processing and ML training to proceed in tandem with the simulation work. In the end the lack of a fully working blackhole simulation of the right size did limit what was achievable with the ML, however it did not prevent proof of concept work.

**Difficulties with setting up working environments should not be underestimated.**

A number of difficulties were encountered in setting up the environment for NS-3 with Linux as described in the risks mitigated section. The initial difficulties related to non-native Linux machines and being able to install all the required dependencies on these. Running simulations on a dedicated remote VM with a native Linux OS would help to mitigate these difficulties. It would also make work sharing on the simulation easier.

Configuring and running NS-3 simulations:

Further problems related to the difference in the NS3 versions between the source scripts that we intended to use and the default installations of the latest NS3 version. Significant changes were made to the NS3 simulation environment between version 3.25 and version 3.30 including removing some modules that were used in the source scripts. These changes substantially increased the amount of work involved in creating the simulation scripts for our scenarios.   
Lesson learned – the initial effort of making sure that you have installed synchronised compatible versions of all the OS and software components that match the versions of the source scripts is worthwhile.

# Handover Materials to the Sponsor

TH & TD to add their materials here.

With a project of this scope, there were many separate scripts developed and output files created. These materials plus accompanying documentation were handed over to the project sponsor. These materials include:

* Project Closure Report.
* Literature Review.
* Jupyter Notebooks Script to convert the network trace files of each node into a dataset.
* Jupyter Notebook Script to create machine learning models to detect black hole nodes from the datasets.
* Excel sheet explaining dataset features.
* Tuned Random Forest and Support Vector Machine models to detect and classify black hole nodes.
* Zip file of example node trace files (pcap) and generated datasets to be used for the machine learning.

Json files are not included due to their immense size. They can easily be created from by loading pcap files into Wireshark and converting them to json files.

# Recommendations to the Sponsor

It has already been well documented that the network simulations were not running as expected according to the AODV protocol. In spite of this, there have been many successes and the project shows much promise that a very accurate solution to detecting and circumventing black hole attacks can be achieved through machine learning techniques. It is recommended that further work be carried out on this project as we feel confident that a good working solution can be achieved.

Many recommendations can be made to the sponsor and to anyone wishing to continue to work on this project. They are explained in the section “Opportunities for Future Development”. However, the recommendations with the highest priorities are briefly mentioned here.

* It is recommended for any future work that the first step is to ensure the NS-3 simulations are working accurately and modelling black hole nodes accurately.
* With accurate simulation trace files, thoroughly test the data conversion and machine learning scripts, to ensure that the scripts are running as expected and that they are able to produce accurate results.
* Larger datasets comprising several simulations with different network configurations should be combined to train the models to detect black hole nodes in various conditions.

# Opportunities for Future Development

There is so much more interesting work that could be done to improve the scope of this project and its effectiveness at nullifying black hole nodes in real life scenarios. This is beyond the scope of a one semester project. Many important steps include:

* Improve the NS-3 simulations so that the simulations are behaving as expected according to the AODV protocol.
* Improve the black hole node simulations. For the 1st stage of simulations, the black hole nodes should be “dumb” where their behaviour is obvious, i.e. they act exactly as is described in the project deliverables document XXXX.
* Once these are working accurately, create several datasets with various network configurations, including:
  + Network size
  + Number of nodes,
  + simulation run time
  + mobility of nodes
  + activity of nodes
  + number of black hole nodes.
* Combine all of these datasets of the various network configurations into one large training dataset. This larger set will used to train the models to various configurations of networks, so that it may more accurately be able to identify black hole nodes in certain network conditions. Blackhole nodes should not be the subject node in the dataset, as it is desired to train normal nodes to detect black hole nodes.
* The test dataset should also be a large and consist of a wide variety of network conditions.
* After the models have been trained, a new network simulation can easily be configured and run and used to detect the model’s accuracy at detecting the black hole nodes.
* After the final models have been trained, the aim is to deploy these algorithms into the normal nodes within the network. These normal nodes will monitor their 1st tier neighbours behaviour for a certain period of time and should be able to detect if a 1st tier neighbour is a black hole node. If so, the subject node should be trained to find alternative routes to the desired destination, effectively avoiding and isolating the black hole node.
* Further work from the previous step could be to tune how much activity data is required for the subject node to accurately detect the black hole node. The idea is to minimise the amount of time to detect the black hole node so that disruption to network traffic is minimised.
* Total network traffic could be monitored for both:
  + 1. normal nodes
  + 2. All nodes made “smart” by incorporating the ML algorithms to detect and bypass black hole nodes

The difference in total network traffic degradation could be monitored and compared from the two scenarios.

* Progressively make the black hole nodes “smarter” so that their behaviour is not consistently so obvious and repeat the network simulations to test the models’ accuracy of detection.
* Write and publish a paper about the project and the methods developed to detect and combat black hole attacks in MANET networks.

# Sponsor Signoff

Ask the sponsor to Complete the “proforma” and ask for a formal signoff and include it in the report. This will be discussed on Thursday 27/04