**Preventing Black Hole**

**Attacks In Manets Using**

**Dynamically Generated Audit Data**

   Final Report

 Group 11522-23-16

|  |  |  |
| --- | --- | --- |
| Authors | Version | Date |
| Thi Doan, Alan Gaugler, Tristan Hore, Johnson Sangah | 1.0 | 21 April 2023 |

     Approval by Team Members:

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

​​

Table of Contents

​

​ To be completed . No executive summary

Contents

[Executive Summary 2](#_Toc577683306)

[Introduction and Description of the Project 3](#_Toc918807581)

[Summary of Outcomes 5](#_Toc653485379)

[Report on the Project Deliverables 5](#_Toc93645781)

[Handover Materials to the Sponsor 16](#_Toc1879008507)

[Opportunities for Future Development 17](#_Toc1660833351)

[Sponsor Signoff 18](#_Toc1158047394)

​

​

​

​

​

​

​

​

​

​

​

​

​

​

​​

# Introduction and Description of the Project

A Mobile Ad Hoc Network (MANET) is a type of wireless network where several devices, such as radios, smartphones, laptops, drones or sensors, communicate directly without needing any pre-existing or centralised infrastructure. MANETs are self-configuring and self-maintained by their nodes, making them ideal for situations where existing network infrastructure is damaged, unavailable, or not possible such as in disaster-stricken areas, military operations, or remote regions.

For example, in a region recently devastated by an earthquake, communication infrastructure such as mobile networks may be severely damaged. In such an event, rescue and emergency services could rapidly set up a MANET to quickly establish a wireless network among their devices to exchange vital information and communications. Likewise, in military operations where fixed network infrastructure is unavailable, MANETs can enable military personnel and unmanned devices to communicate and share tactical information in real-time using wireless devices.

A commonly used routing protocol in MANETs is the Ad-hoc On-Demand Distance Vector (AODV) protocol. AODV is a reactive routing protocol, which means that it establishes routes between nodes on-demand when required rather than maintaining pre-established routes. AODV uses a route discovery mechanism where nodes broadcast route requests to find a route to a destination node. When a route is discovered, AODV establishes a temporary route and maintains it for as long as the nodes are actively communicating. AODV protocol is widely used in MANETs due to its simplicity, scalability, and adaptability to changing network topologies due to the mobility of nodes.

However, protocols like AODV are vulnerable to various types of cyber-security threats, one of the most common being black hole attacks. A black hole attack is a type of cyber-attack where a malicious node infiltrates the network and falsely claims to have the shortest route to a destination, making it the prioritised path for link establishment. Doing so attracts routing and data traffic towards it, but it drops the received packets instead of forwarding them to the desired destination. This will severely disrupt communication in the network between nodes and cause the loss of vital data, leading to performance degradation and network instability. Hence, black hole attacks can cause severe disruptions in critical operations, making them a significant security concern in MANET environments.

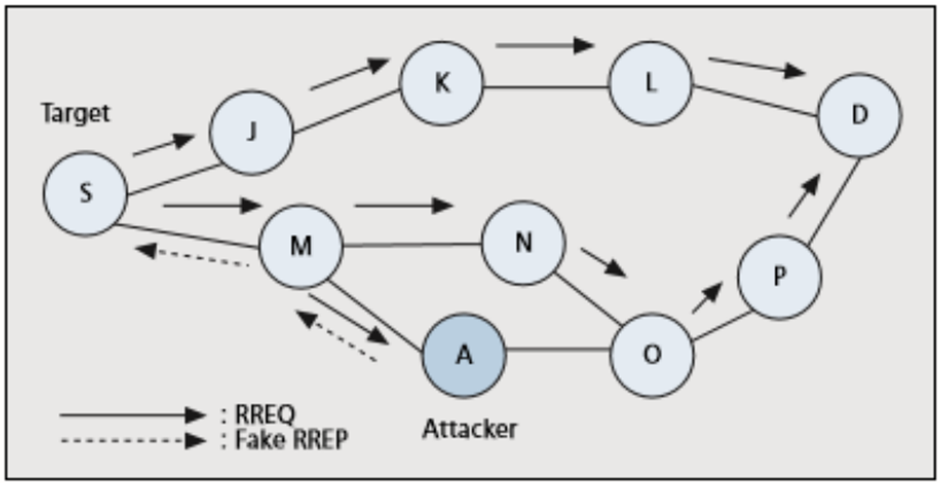


Figure 1 – Diagram of MANET with a Black Hole Node A (Source: Fakultas et al.)

Figure 1 shows a diagram of a MANET where the Source Node S sends a route request to the Destination Node D. The malicious Node A replies with a fake Route Response message, claiming to have the shortest route to Node D. Node S would then establish a route to Node D through Node A, but the packets would be dropped by Node A instead of being forwarded to the destination.

This project aimed to develop a solution using machine learning techniques that could detect black hole intruders in a MANET utilising AODV protocol. The project consisted of three major phases.

In the Data creation phase, a MANET network would be simulated using the NS3 simulator, incorporating the AODV routing protocol. The simulation would also include a small number of black hole nodes. Finally, the simulation would generate output files comprising various network metrics, AODV messaging and data flow between the nodes.

In the Data preparation phase, a script would be developed to process the simulation files and convert them into a dataset that could be used to train a machine learning classifier. This would involve identifying and extracting various relevant features that could be used to detect the behaviour of black hole nodes.

The machine learning (ML) phase would develop a machine learning process using the dataset generated in Stage 2 as input to train ML classifiers to detect which neighbour nodes were black hole nodes.

# **Summary of Outcomes**

**Data Creation Phase:** An NS3 simulator was used to simulate MANET networks using the AODV protocol. Three scripts were developed in C+ to simulate the interaction of all nodes in the MANET. Black hole nodes were also introduced into the network. The scripts were able to produce NS3 simulations of normal AODV networks. However, due to the complexity of the process, the scripts were producing some inaccuracies in the AODV with black hole node simulations. Despite this, the scripts produced good output that could still be used for the project's next phases.

**Data preparation phase:** The literature review identified the key behavioural characteristics of black hole nodes in AODV networks. From this, a complex program was developed that would read the AODV protocol messaging between nodes from the trace files produced in the previous phase. The script would then extract key features from the messaging from which a dataset was created that could be used in the machine learning process. This phase was successful, and datasets could be produced from the trace files.

Due to issues with the NS-3 simulation, there were some inaccuracies in the simulation of the AODV protocol and the black hole nodes. As a result, some values of the feature set of the black hole nodes were manually altered in the training set to accurately portray the behavioural characteristics of a black hole node. With this step, the project could continue.

**Machine Learning Phase:** A machine learning script was developed to detect black hole nodes within the network. The datasets created by the data conversion script developed in the Data Preparation Stage were used as the input. The variables with the highest correlation to the target variable were determined and they were filtered into the final dataset for model training. Two binary classification models were developed, a random forest classifier and a support vector machine classifier. Both models could classify with 100% accuracy which nodes were black hole nodes.

More work must be done to get the NS-3 simulator to accurately simulate AODV networks and black hole nodes and then more training and testing of the models is required. The future work section mentions what steps should be carried out next. However, with all the hard work put into this project, the project's final phase was successfully completed. The results demonstrate that machine learning can successfully be applied to AODV networks to detect black hole nodes accurately.

# Report on the Project Deliverables

In the initial scope of the project, six major deliverables were defined. This section reports on the processes and outcomes of each of these deliverables.

1. **Literature review**

The team carried out 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. This in-depth review was essential in correctly understanding how nodes in MANETs, 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, disrupting essential communications.

The team also viewed many videos, which proved essential in learning the protocols and configuration of the NS-3 network.

The literature review gave us a solid understanding of MANETs, the AODV network protocol, and the behavioural characteristics of the black hole nodes infiltrating the network. Several of these characteristics could be used for a network node to identify a first-tier neighbour as a black hole node.

Identifying these principal 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 of the identified behavioural characteristics of black hole nodes and for a bibliography of the most informative and relevant research papers reviewed for this project.

1. **Source code for the NS-3 simulations**

The Network Simulator 3 (NS3), is a powerful, open-source network simulation tool developed by the NS3 community. Originating in 2006 as an evolution of the NS-2 simulator, NS3 provides a comprehensive platform for researchers, and engineers to design, prototype, and analyze networks. NS3's modular architecture supports various network protocols, traffic models, and routing algorithms, this enables users to simulate complex, large-scale networks. The simulator is widely used in academia and industry to study wireless, wired, and hybrid networks and test and validate new networking technologies.

In this project three scripts modified from the standard MS3 modules in line with existing blackhole simultation examples, were used in the preparation of the NS3 simulation, along with many standard NS3 modules and tools. In addition, a simple blackhole model was used early on to understand network behaviour, providing a foundation for anticipating the behaviour in a more extensive network. [REF]

Diagram

Description automatically generated

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 several issues and required substantial debugging in addition to the changes needed to implement the blackhole simulations. In addition, further changes were required to match the changes in the AODV-Routing-Protocol.cc and AODV-Routing-Protocol.h scripts 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]

1. **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 helper 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. Finally, 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 primary 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.

1. **Dataset Preparation**

The next phase of the process was to create a dataset that could be used for the machine learning phase. 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 sufficient 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 loaded into Wireshark, where they could be observed. Wireshark is a free and open-source network protocol analyser.

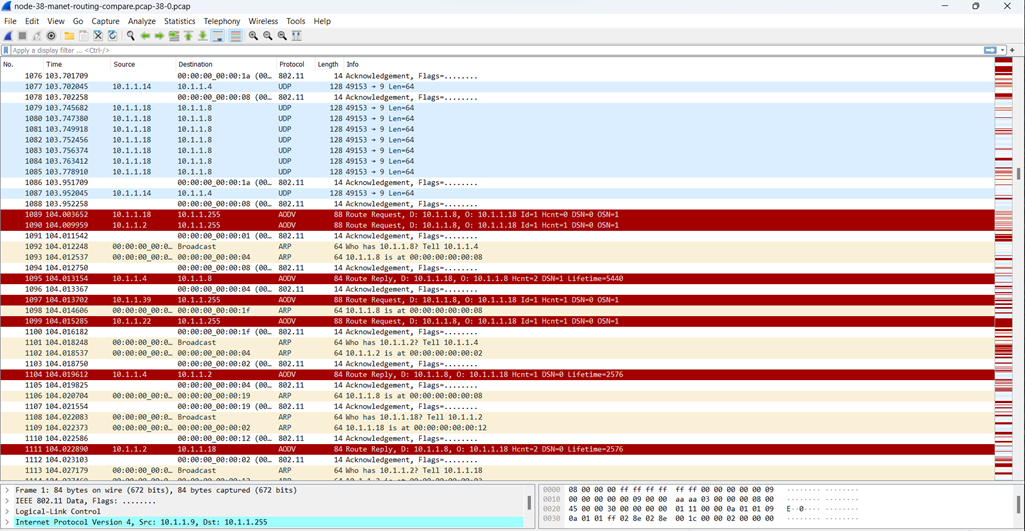


Figure 1 – Example display of AODV messaging in Wireshark

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

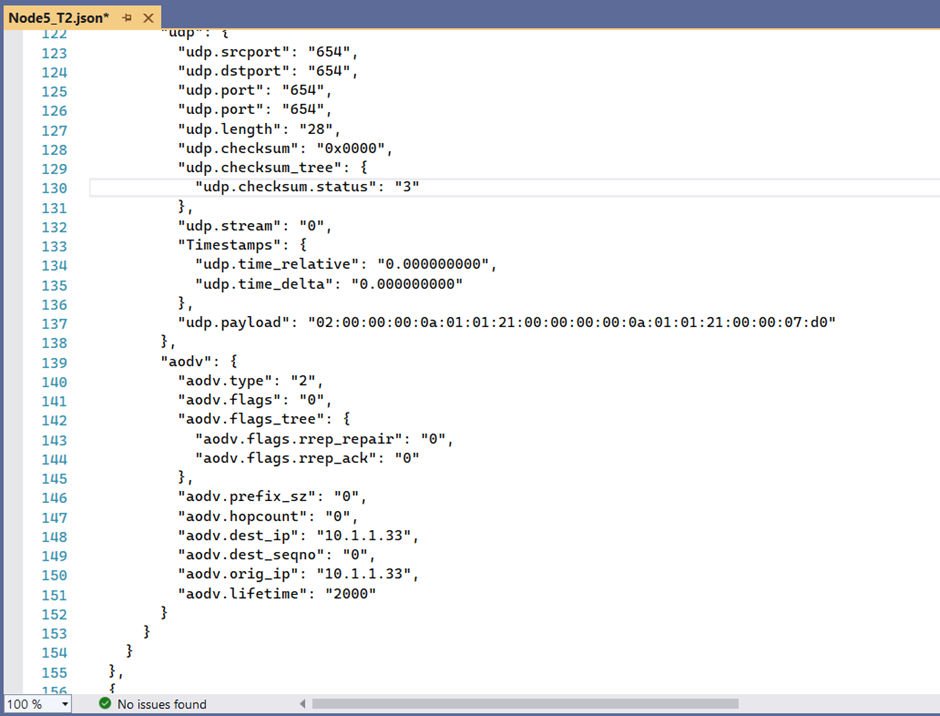


Figure 2 –Example of AODV messaging in JSON format view in Visual Studio

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.

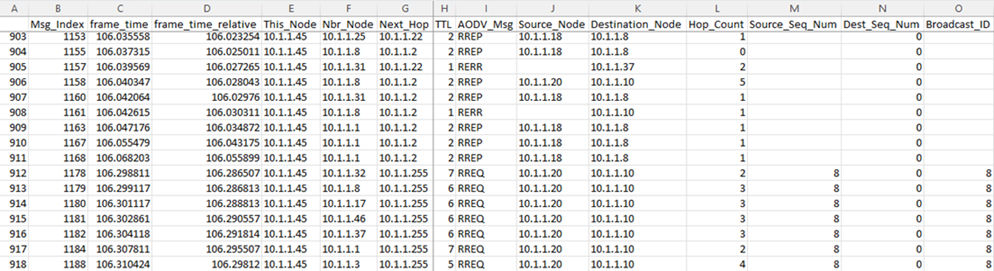


Figure 3 – Example Output of a Data Frame after Stage 1

**Stage 2**

Some processing is made for each AODV message, creating new features in the data frame: 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.

Suppose the message is a RREP message responding to a RREQ message. In that case, 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.

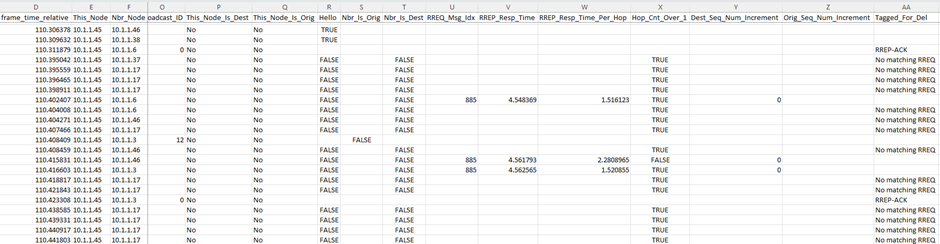


Figure 4 – Example Output of Additional Features of a Data Frame added after Stage 2

**Stage 3**

This stage will convert the data frames from Stage 2 into datasets for each Subject Node and then merge them all into one combined dataset. Every row represents features between the subject node and each of its first 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.

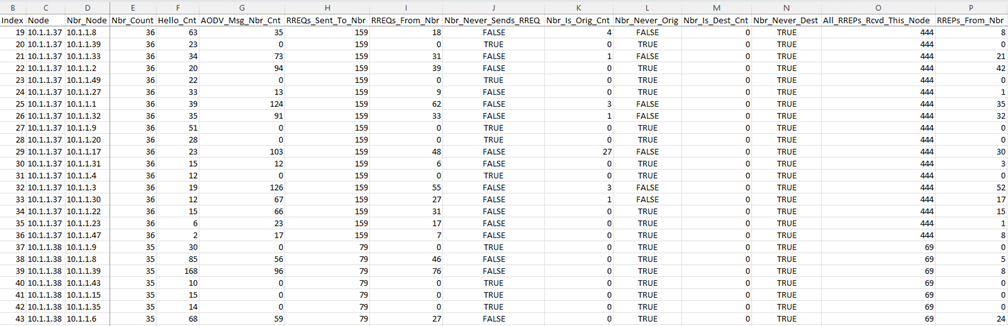


Figure 5 – Example Output of the Dataset produced after Stage 3

**Black Hole Node Data Alteration**

It must be stressed that black hole nodes simulations were not able to be implemented in the NS-3 simulator within the timeframe of this project, so the values in the datasets have been manually 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 hole’s behaviour is considered “typical” or “dumb”, and its characteristics are extremely obvious. This would be the typical behaviour of a black hole node that is not expecting the network nodes to have any intelligence in detecting and mitigating malicious nodes, which is true for most real black hole attacks.

As an example of the data that was altered for black hole nodes, the variable RREQs\_From\_Nbr was always set to 0 and its corresponding flag feature Nbr\_Never\_Sends\_RREQ was always set to True. This is the behaviour that is expected from a black hole as it would always be responding to an RREQ (Route Request) with a RREP (Route Reply) claiming to have the shortest route to the destination, so it would never broadcast the RREQ further on to its neighbours resulting in a count of 0. Likewise, RREP\_Resp\_Pct (RREP response percentage) would always be 100% because the Black Hole Node always responds to any RREQ it receives.

For a more detailed explanation of what values were manually altered please refer to the end of the “Dataset Creation” section of the Project Closure Report.

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

1. **Machine Learning Process**

A machine learning script was developed using Python and Jupyter Notebooks 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.

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.

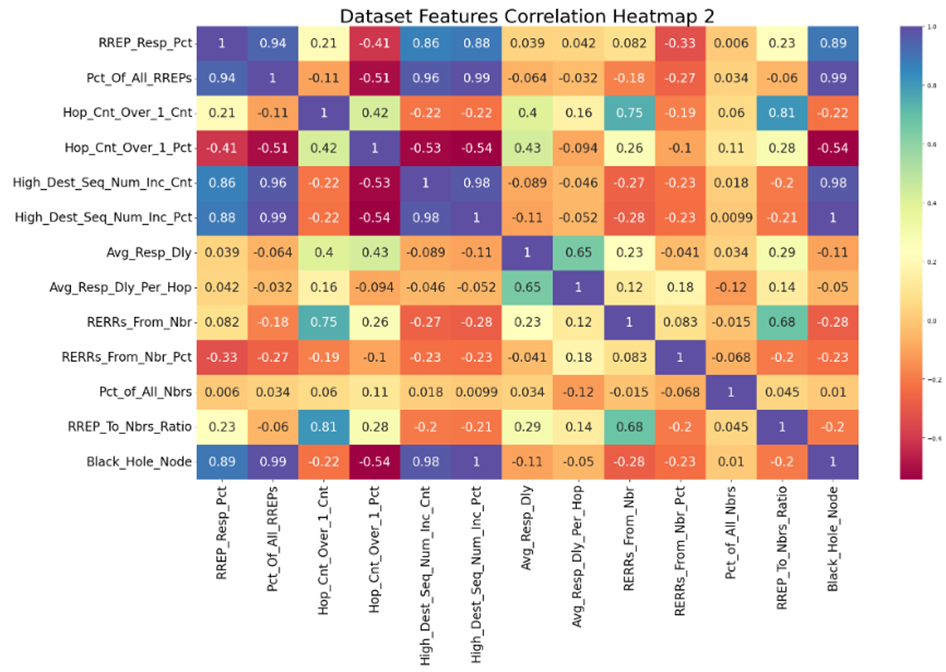


Figure 6 – Example output of the correlation heatmap demonstrating the relationship between the features and the target variable (bottom row)

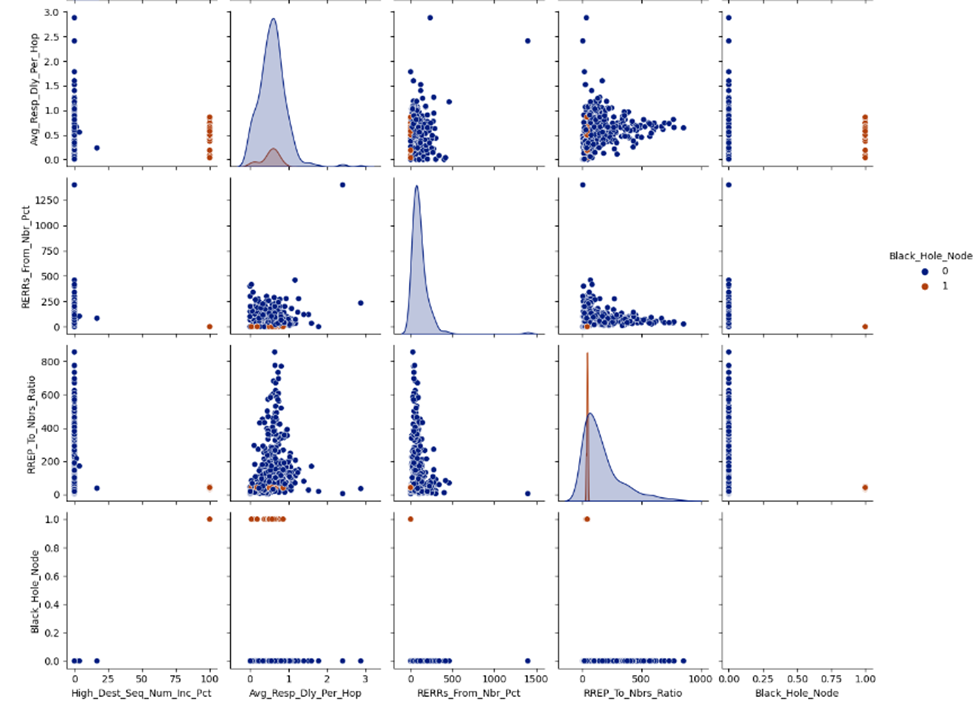


Figure 7 – Example Output of Pair Plots showing the relationships between variables.

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. Standardisation 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%.

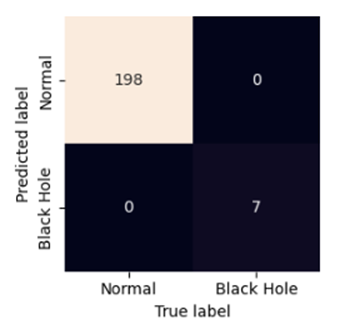


Figure 8 – Example of the Confusion Matrix from the SVM Model Classifying Black Hole Nodes.

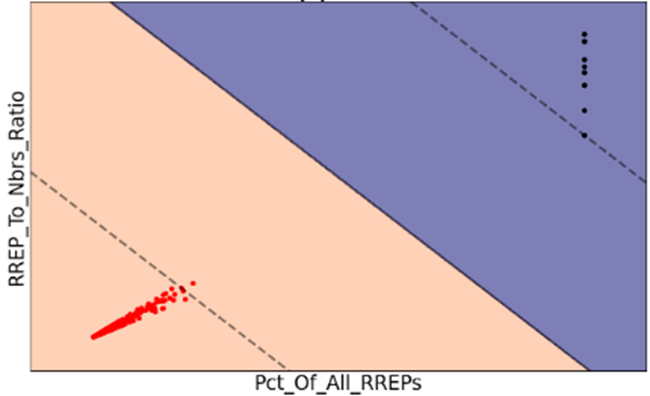


Figure 9 – Example of SVM Visualization of support vectors. There is a large separation between normal nodes (red) and black hole nodes (black).

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”.

1. **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 generate, convert and remodel the data.

**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**

**NS-3 Network Simulations**

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 with the simulations of AODV network behaviour. Most of these were issues with the messaging and protocol behaviour not strictly following AODV protocol.

**NS-3 Black Hole Node Simulations**

Another important issue was the accurate simulation black hole nodes. Unfortunately, the duration of the project did not permit enough time to correct these issues. As mentioned, the

**Data Conversion and Machine Learning Processes and Scripts**

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.

Despite these issues, we were able to engineer a solution for the project to progress and successfully complete the machine learning stage.

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

**Report of Risks Mitigated**

**Format this section to landscape. I can’t do it in MS Teams**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Risk | Risk Description | Likelihood | Impact | Status | Mitigation |
| Personnel Issues | It is possible that any project member could become sick or may have to leave the project due to personal issues. | Moderate | Moderate | Occurred | One member became sick for a few weeks.  Overlapping responsibilities with other members.  Two members were simultaneously working on the NS-3 simulations. |
| Technical Difficulties | Difficulties with configuring the NS-3 environment and building and running the scripts. | Moderate | Severe | Occurred | Several difficulties with configuring the NS-3 environment and building and running the scripts.  Two different NS-3 environments were setup on two different machines. The enabled the team to progress further with the project.  Output datasets were altered to accurately simulate black hole node behaviour. |
| Hardware Requirements | Personal Computers may not possess sufficient processing power. | Low | Low | Did Not Occur | AWS remote environment will be utilised for the NS-3 simulation. |
| Software Configuration | Issues with running Python scripts or other required software on personal PCs. | Low | Low | Did Not Occur | Using online platforms such as Google Colab for the ML training. |
| Hardware Issues | A PC used on the project may be damaged or stolen. | Moderate | Moderate | Did Not Occur | 1. Files will be checked out of team GitHub.  2. Word documents will be edited on the team’s online file repository.   3. Other files should be stored on the team’s online file repository. |
| Unforeseen Project Delays | Project delays if there are unresolved issues that impede progress along the way. This could result in additional costs if the project were to be delayed. | Moderate | Severe | Did Not Occur | The project scope will be clearly defined before beginning and ensure that the entire project can be completed within the designated timeframe. By doing so, we aim to prevent potential delays and minimise any associated additional costs that could arise due to project overruns. |
| Unrealistic Expectations | Unclear or Unrealistic Expectations can arise when there is a lack of mutual understanding among all parties involved in the project. Unrealistic expectations may lead to unattainable project goals or lower quality standards. | Moderate | Moderate | Did Not Occur | A careful planning process, including the estimation of costs and time required to complete milestones. Additionally, effective communication will be established to ensure that the entire team has a clear understanding of the project's objectives and expectations. |
| Not Meeting Deadlines | Missing Deadlines and Deliverables. | Low | Severe | Did Not Occur | Prioritise tasks by allocating more resources to those that are crucial to the project and giving them the highest priority. Conversely, low-priority tasks will receive fewer resources. Identification of dependencies and completion of tasks synchronously whenever possible. Implementing these measures will ensure that all critical deadlines and deliverables are met on time. |
| Poor Communication | Poor communication among team members and stakeholders. | Moderate | Severe | Did Not Occur | Stakeholders will be informed of the most effective communication channels. Team members will meet regularly and provide weekly updates to the project manager on the progress of their assigned tasks. Sponsors will receive regular progress reports. This will ensure that all team members are well-informed and that there is no miscommunication. By implementing this strategy, communication will improve, and this will facilitate a better understanding of the project's progress among all stakeholders. |

**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 time frame for this project and many challenging technical tasks, one serious technical hurdle could likely cause lengthy delays and ensure that 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 successfully generated simulation files that 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 were generated from scripts, so it is 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, being ethical and honest about any changes we have made is always imperative. 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

With a project of this scope, many separate scripts were 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 that were used for the machine learning phase.
* Modified AODV routing protocol modules
* Modified Manet routing compare script
* Notes on NS3 Version installation and environment setup.

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.

To view all the handover materials, please refer to our GitHub repository.

[11522-23-16 Machine Learning for Black Hole Detection](https://github.com/capstone11522manet/manet)

**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, the project overall has been a success and it holds much potential 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.

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

* For future work, the first step is to ensure the NS-3 simulations are working 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 can 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

Many exciting extensions to this work could be done to improve the applicability 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. Several essential steps include:

* Improve the NS-3 simulations so that the simulations with blackhole nodes are behaving as expected according to the AODV protocol.
* Improve the black hole node simulations. For the first 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 Closure Report.
* 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 the various network configuration datasets into one large training dataset. This larger set will be used to train the models to various configurations of networks, so that it may more accurately 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 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, they can be deployed into the normal nodes within the network. These normal nodes will monitor their first-tier neighbours’ behaviour for a certain period and should be able to detect if a first-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 detect the black hole nodes accurately. The idea is to minimise the amount of time to detect the black hole node so that disruption to network traffic is minimised.
* A further step could be to use the NS3-AI module to deploy ML models in the simulated nodes in MANET, to mitigate the effects of a black hole 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 black hole attack could be determined in comparisons between simulations with and without deployed models.
* 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 detection rate.
* 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