Project Closure Report for

Preventing Black Hole Attacks in

MANETS Using Machine Learning

Audience
Project sponsors, project mentors, unit coordinators and all others in the university and ITC community interested in this project.

Purpose

The purpose of this document is to summarise the work completed in this project. clarify the scope, methodology and objectives of this capstone project. The main stakeholders should start with the closure report and proceed to read the final report.

Authors	Version	Date
Thi Doan, Alan Gaugler, Tristan Hore, Johnson Sangah	1.0	3 May 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

NS3 Simulation

Resources Used for NS3 Simulation	Resources Used for Data Analysis and Machine Learning
Network Simulator 3 (NS3) – NS 3.37	Python 3.10
Host OS WSL2 on Windows 11	Jupyter Notebooks
Linux OS Ubuntu 20.04.5	Pandas
Python 3.8.10	Scikit-Learn
	Matplotlib
	Seaborn

There were several issues with this setup are missing bindings between the NS3 version and the Python version. Following the setup steps in the setup tutorial for NS3 mostly worked, some issues were found with changes in components between the tutorial and the current version. The installation method described is now deprecated in newer versions and caused some issues. The newest installation method was not described in detail and did not work well on WSL2 or the MacOS in our experience.

The source script described below was created for the NS3.25 version and a number of changes in the modules and backward compatibility for the more recent versions of NS3 require changes to this script. A file manet-routing-compare.cc with these changes is included in the NS3 files of the handover material.

Description of the scripts

The original Mobile Adhoc Network (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 and released under a GNU General Public Licence. This part of the project aimed to modify this script to be able to compare different implementations of AODV with and without blackhole attacks. The modifications create the code to allow for a Blackhole attack simulation by creating malicious nodes using the routing protocol. A Blackhole attack is a type of network security breach in which a malicious node advertises itself as having the shortest path to the destination node in the network, causing other nodes to route their packets through the malicious node. The malicious node then drops the received packets, disrupting the network communication.

The simulation relies on changes based on the work of Shalini Satre and Mohit P. Tahiliani. *(Mohit P. Tahiliani: [Ns-3] Blackhole Attack Simulation in Ns-3, n.d.)*These changes were modifications of the original Ad hoc On-Demand Distance Vector (AODV) routing protocol implementation in the NS-3 network simulator as described below. Copies of the modified files are included in the handover material.

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.

Description of the function of the AODV routing files

The AODV routing files are standard NS3 module components that implement the AODV routing protocol in this software package. We use these as the basis for the modified protocol with black hole node behaviour implimented.

Description of the modifications to the code in AODV files

This script is a modification of the original AODV 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 modification introduces a Blackhole attack simulation. A Blackhole attack is a type of network security breach in which a malicious node advertises itself as having the shortest path to the destination node in the network, causing other nodes to route their packets through the malicious node. The malicious node then drops the received packets, disrupting the network communication.

The modified implementation adds a new attribute, "IsMalicious", to the AODV routing protocol class. The attribute is used to mark a node as malicious. Two new methods, SetMaliciousEnable() and GetMaliciousEnable(), are also added to set and retrieve the "IsMalicious" attribute value. In the modified implementation, two changes are made to the behavior of the AODV routing protocol when a node is marked as malicious:

Firstly, when a malicious node receives a packet, it simply drops the packet instead of forwarding it. Secondly, 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 AODV network, allowing us to study the impact of such attacks on network performance and develop countermeasures to protect the network.

Description of the Manet routing compare file

This is an ns-3 simulation script that compares the performance of three different MANET routing protocols: Optimized Link State Routing (OLSR), Ad hoc On-Demand Distance Vector (AODV), and Destination-Sequenced Distance Vector (DSDV) in a random waypoint mobility model. The simulation is configurable to adjust the number of nodes, node speed, mobility model, transmission power, and other parameters. All nodes are both sources and sinks.

The script sets up a wireless network simulation with the number of nodes configured, simulates their movement using the Random Waypoint Mobility Model, and evaluates the performance of the chosen routing protocol (OLSR, AODV, or DSDV) in terms of the number of received packets. The results are saved to a CSV file for further analysis. The CSV file, contains information about the simulation time, receive rate, packets received, number of sinks, routing protocol, and transmission power. The script also outputs the routing message logs to both a flowmon file and to PCAP files. The Flomon file is a single xml format file for all nodes.

Description of the modifications to the code in the manet routing compare file

In this script modification, we add logging for The PCAP files are output separately for each node. Issues with deprecated modules used in the original script(which was for NS3 version 3.25) are addressed and the latest WIFI model is used. Code to flag random nodes in the range of nodes created as Malicious is implemented – ***NB this section of code is currently not working correctly and has been commented out.***

Code to output the Malicious attribue flags to the file names is implimented – ***NB this section of code is currently not working correctly and has been commented out.***
Further comments are included in the surce code of the scripts.

Dataset Creation

The main behavioural characteristics of black hole nodes in AODV 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 the machine learning models is not an issue. With proper simulation files available It is recommended to run several simulations with different configurations and merge them to create the training set (ensure there are no duplicate Node ID names) and this can also be done for the test set. Once the model has been tuned, this can be applied to an individual network simulation and used to detect the black hole nodes within it.

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

https://www.wireshark.org

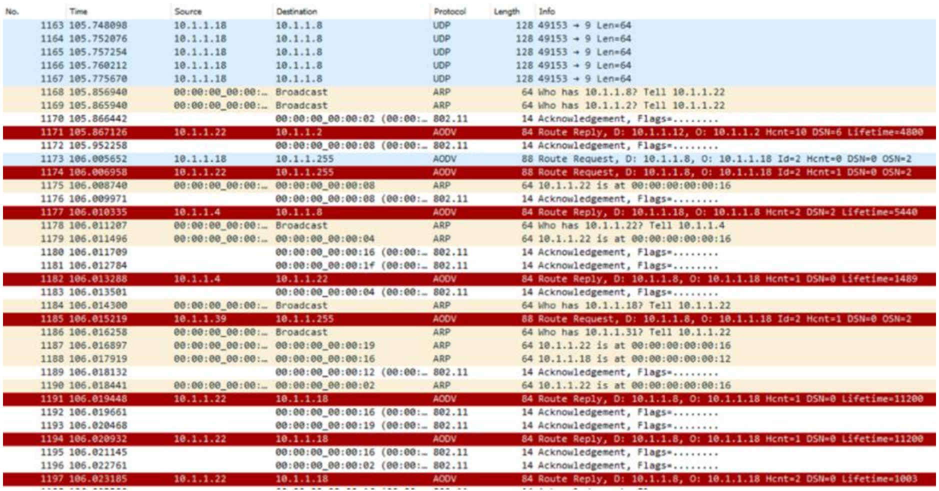


Figure 1 – Example display of AODV messaging in Wireshark

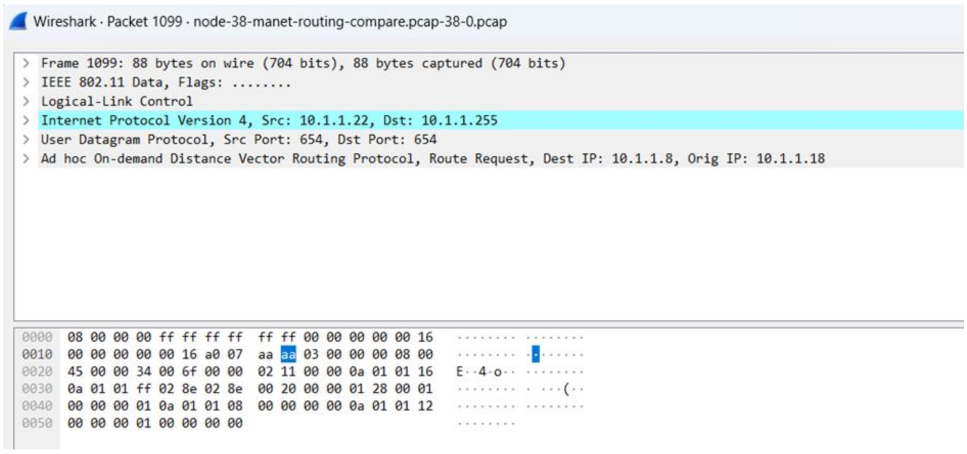
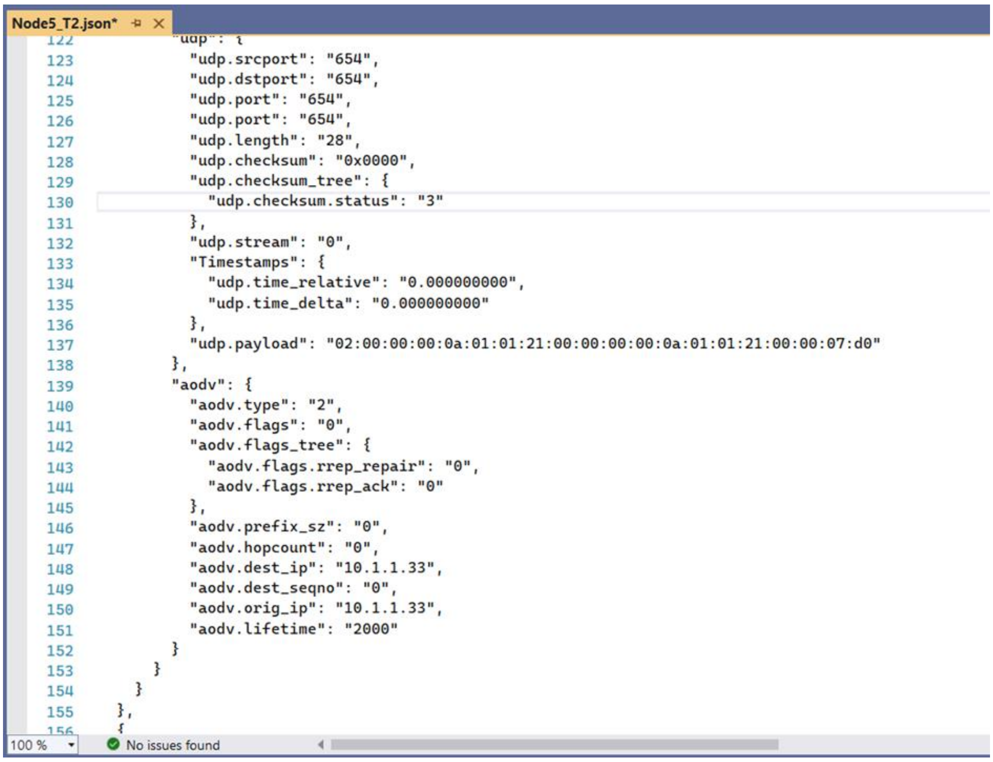


Figure 2 –AODV message details in Wireshark

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




```
Node5_T2.json*
        "udp.srcport": "654",
        "udp.dstport": "654",
        "udp.port": "654",
        "udp.port": "654",
        "udp.length": "28",
        "udp.checksum": "0x0000",
        "udp.checksum_tree": {
          "udp.checksum.status": "3"
        },
        "udp.stream": "0",
        "Timestamps": {
          "udp.time_relative": "0.000000000",
          "udp.time_delta": "0.000000000"
        },
        "udp.payload": "02:00:00:00:0a:01:01:21:00:00:00:00:0a:01:01:21:00:00:07:d0"
      },
      "aodv": {
        "aodv.type": "2",
        "aodv.flags": "0",
        "aodv.flags_tree": {
          "aodv.flags.rrep_repair": "0",
          "aodv.flags.rrep_ack": "0"
        },
        "aodv.prefix_sz": "0",
        "aodv.hopcount": "0",
        "aodv.dest_ip": "10.1.1.33",
        "aodv.dest_seqno": "0",
        "aodv.orig_ip": "10.1.1.33",
        "aodv.lifetime": "2000"
      }
    }
  }
},
100 %    No issues found
```


Figure 3 –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. This script is included in the project deliverables with the name "AODV_to_Dataset.ipynb".

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.

The process of converting the individual node trace files into a usable dataset 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_2.csv files, each row refers to each extracted message marked by a frame time. It contains information on 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.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	Msg_Index	frame_time	frame_time_relative	This_Node	Nbr_Node	Next_Hop	TTL	AODV_Msg	Source_Node	Destination_Node	Hop_Count	Source_Seq_Num	Dest_Seq_Num	Broadcast_ID
903	1153	106.035558	106.023254	10.1.1.45	10.1.1.25	10.1.1.22	2	RREP	10.1.1.18	10.1.1.8	1		0	
904	1155	106.037315	106.025011	10.1.1.45	10.1.1.8	10.1.1.2	2	RREP	10.1.1.18	10.1.1.8	0		0	
905	1157	106.039569	106.027265	10.1.1.45	10.1.1.31	10.1.1.22	1	RERR		10.1.1.37	2		0	
906	1158	106.040347	106.028043	10.1.1.45	10.1.1.8	10.1.1.2	2	RREP	10.1.1.20	10.1.1.10	5		0	
907	1160	106.042064	106.02976	10.1.1.45	10.1.1.31	10.1.1.2	2	RREP	10.1.1.18	10.1.1.8	1		0	
908	1161	106.042615	106.030311	10.1.1.45	10.1.1.8	10.1.1.2	1	RERR		10.1.1.10	1		0	
909	1163	106.047176	106.034872	10.1.1.45	10.1.1.1	10.1.1.2	2	RREP	10.1.1.18	10.1.1.8	1		0	
910	1167	106.055479	106.043175	10.1.1.45	10.1.1.1	10.1.1.2	2	RREP	10.1.1.18	10.1.1.8	1		0	
911	1168	106.068203	106.055899	10.1.1.45	10.1.1.1	10.1.1.2	2	RREP	10.1.1.18	10.1.1.8	1		0	
912	1178	106.298811	106.286507	10.1.1.45	10.1.1.32	10.1.1.255	7	RREQ	10.1.1.20	10.1.1.10	2	8	0	8
913	1179	106.299117	106.286813	10.1.1.45	10.1.1.8	10.1.1.255	6	RREQ	10.1.1.20	10.1.1.10	3	8	0	8
914	1180	106.301117	106.288813	10.1.1.45	10.1.1.17	10.1.1.255	6	RREQ	10.1.1.20	10.1.1.10	3	8	0	8
915	1181	106.302861	106.290557	10.1.1.45	10.1.1.46	10.1.1.255	6	RREQ	10.1.1.20	10.1.1.10	3	8	0	8
916	1182	106.304118	106.291814	10.1.1.45	10.1.1.37	10.1.1.255	6	RREQ	10.1.1.20	10.1.1.10	3	8	0	8
917	1184	106.307811	106.295507	10.1.1.45	10.1.1.1	10.1.1.255	7	RREQ	10.1.1.20	10.1.1.10	2	8	0	8
918	1188	106.310424	106.29812	10.1.1.45	10.1.1.3	10.1.1.255	5	RREQ	10.1.1.20	10.1.1.10	4	8	0	8

Figure 4 – 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 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.

D	E	F	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA
frame_time_relative	This_Node	Nbr_Node	oadcast_ID	This_Node_Is_Dest	This_Node_Is_Orig	Hello	Nbr_Is_Orig	Nbr_Is_Dest	RREQ_Msg_Idx	RREP_Resp_Time	RREP_Resp_Time_Per_Hop	Hop_Cnt_Over_1	Dest_Seq_Num_Increment	Orig_Seq_Num_Increment	Tagged_For_Del
110.306378	10.1.1.45	10.1.1.46		No	No	TRUE									
110.309632	10.1.1.45	10.1.1.38		No	No	TRUE									
110.311879	10.1.1.45	10.1.1.6	0	No	No										RREP-ACK
110.395042	10.1.1.45	10.1.1.37		No	No	FALSE		FALSE				TRUE			No matching RREQ
110.395559	10.1.1.45	10.1.1.17		No	No	FALSE		FALSE				TRUE			No matching RREQ
110.396465	10.1.1.45	10.1.1.17		No	No	FALSE		FALSE				TRUE			No matching RREQ
110.398911	10.1.1.45	10.1.1.17		No	No	FALSE		FALSE				TRUE			No matching RREQ
110.402407	10.1.1.45	10.1.1.6		No	No	FALSE		FALSE	885	4.548369	1.516123	TRUE	0		
110.404008	10.1.1.45	10.1.1.6		No	No	FALSE		FALSE				TRUE			No matching RREQ
110.404271	10.1.1.45	10.1.1.46		No	No	FALSE		FALSE				TRUE			No matching RREQ
110.407466	10.1.1.45	10.1.1.17		No	No	FALSE		FALSE				TRUE			No matching RREQ
110.408409	10.1.1.45	10.1.1.3	12	No	No		FALSE								
110.408459	10.1.1.45	10.1.1.46		No	No	FALSE		FALSE				TRUE			No matching RREQ
110.415831	10.1.1.45	10.1.1.46		No	No	FALSE		FALSE	885	4.561793	2.2808965	FALSE	0		
110.416603	10.1.1.45	10.1.1.3		No	No	FALSE		FALSE	885	4.562565	1.520855	TRUE	0		
110.418817	10.1.1.45	10.1.1.17		No	No	FALSE		FALSE				TRUE			No matching RREQ
110.421843	10.1.1.45	10.1.1.17		No	No	FALSE		FALSE				TRUE			No matching RREQ
110.423308	10.1.1.45	10.1.1.3	0	No	No										RREP-ACK
110.438585	10.1.1.45	10.1.1.17		No	No	FALSE		FALSE				TRUE			No matching RREQ
110.439331	10.1.1.45	10.1.1.17		No	No	FALSE		FALSE				TRUE			No matching RREQ
110.440917	10.1.1.45	10.1.1.17		No	No	FALSE		FALSE				TRUE			No matching RREQ
110.441803	10.1.1.45	10.1.1.17		No	No	FALSE		FALSE				TRUE			No matching RREQ

Figure 5 – 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 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.

B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
Index	Node	Nbr_Node	Nbr_Count	Hello_Cnt	AODV_Msg_Nbr_Cnt	RREQs_Sent_To_Nbr	RREQs_From_Nbr	Nbr_Never_Sends_RREQ	Nbr_Is_Orig_Cnt	Nbr_Never_Orig	Nbr_Is_Dest_Cnt	Nbr_Never_Dest	All_RREPs_Rcvd_This_Node	RREPs_From_Nbr
19	10.1.1.37	10.1.1.8	36	63	35	159	18	FALSE	4	FALSE	0	TRUE	444	8
20	10.1.1.37	10.1.1.39	36	23	0	159	0	TRUE	0	TRUE	0	TRUE	444	0
21	10.1.1.37	10.1.1.33	36	34	73	159	31	FALSE	1	FALSE	0	TRUE	444	21
22	10.1.1.37	10.1.1.2	36	20	94	159	39	FALSE	0	TRUE	0	TRUE	444	42
23	10.1.1.37	10.1.1.49	36	22	0	159	0	TRUE	0	TRUE	0	TRUE	444	0
24	10.1.1.37	10.1.1.27	36	33	13	159	9	FALSE	0	TRUE	0	TRUE	444	1
25	10.1.1.37	10.1.1.1	36	39	124	159	62	FALSE	3	FALSE	0	TRUE	444	35
26	10.1.1.37	10.1.1.32	36	35	91	159	33	FALSE	1	FALSE	0	TRUE	444	32
27	10.1.1.37	10.1.1.9	36	51	0	159	0	TRUE	0	TRUE	0	TRUE	444	0
28	10.1.1.37	10.1.1.20	36	28	0	159	0	TRUE	0	TRUE	0	TRUE	444	0
29	10.1.1.37	10.1.1.17	36	23	103	159	48	FALSE	27	FALSE	0	TRUE	444	30
30	10.1.1.37	10.1.1.31	36	15	12	159	6	FALSE	0	TRUE	0	TRUE	444	3
31	10.1.1.37	10.1.1.4	36	12	0	159	0	TRUE	0	TRUE	0	TRUE	444	0
32	10.1.1.37	10.1.1.3	36	19	126	159	55	FALSE	3	FALSE	0	TRUE	444	52
33	10.1.1.37	10.1.1.30	36	12	67	159	27	FALSE	1	FALSE	0	TRUE	444	17
34	10.1.1.37	10.1.1.22	36	15	66	159	31	FALSE	0	TRUE	0	TRUE	444	15
35	10.1.1.37	10.1.1.23	36	6	23	159	17	FALSE	0	TRUE	0	TRUE	444	1
36	10.1.1.37	10.1.1.47	36	2	17	159	7	FALSE	0	TRUE	0	TRUE	444	8
37	10.1.1.38	10.1.1.9	35	30	0	79	0	TRUE	0	TRUE	0	TRUE	69	0
38	10.1.1.38	10.1.1.8	35	85	56	79	46	FALSE	0	TRUE	0	TRUE	69	5
39	10.1.1.38	10.1.1.39	35	168	96	79	76	FALSE	0	TRUE	0	TRUE	69	8
40	10.1.1.38	10.1.1.43	35	10	0	79	0	TRUE	0	TRUE	0	TRUE	69	0
41	10.1.1.38	10.1.1.15	35	15	0	79	0	TRUE	0	TRUE	0	TRUE	69	0
42	10.1.1.38	10.1.1.35	35	14	0	79	0	TRUE	0	TRUE	0	TRUE	69	0
43	10.1.1.38	10.1.1.6	35	68	59	79	27	FALSE	0	TRUE	0	TRUE	69	24

Figure 6 – Example Output of the Dataset produced after Stage 3

B	C	D	U	V	W	X	Y	Z	AA	AB	AC
Index	Node	Nbr_Node	High_Dest_Seq_Num_Inc_Cnt	High_Dest_Seq_Num_Inc_Pct	Avg_Resp_Dly	Avg_Resp_Dly_Per_Hop	RERRs_From_Nbr	RERRs_From_Nbr_Pct	Pct_of_All_Nbrs	RREP_To_Nbrs_Ratio	Black_Hole_Node
16	10.1.1.37	10.1.1.6	0	0	2.801040394	0.886591943	11	33.33	2.78	2.672661871	FALSE
17	10.1.1.37	10.1.1.10	0				0		2.78	0	FALSE
18	10.1.1.37	10.1.1.14	0	0	0.338410333	0.113953889	4	133.33	2.78	0.244604317	FALSE
19	10.1.1.37	10.1.1.8	0	0	1.637109125	0.331291277	9	112.5	2.78	0.647482014	FALSE
20	10.1.1.37	10.1.1.39	0				0		2.78	0	FALSE
21	10.1.1.37	10.1.1.33	0	0	1.829072857	0.433267432	21	100	2.78	1.701438849	FALSE
22	10.1.1.37	10.1.1.2	0	0	3.790838571	1.017852949	13	30.95	2.78	3.402877698	FALSE
23	10.1.1.37	10.1.1.49	159	100			0	0	2.78	35.97122302	TRUE
24	10.1.1.37	10.1.1.27	0	0	0.019969	0.006656333	3	300	2.78	0.082733813	FALSE
25	10.1.1.37	10.1.1.1	0	0	2.103406229	0.804372383	27	77.14	2.78	2.834532374	FALSE
26	10.1.1.37	10.1.1.32	0	0	1.832151781	0.476360634	26	81.25	2.78	2.59352518	FALSE
27	10.1.1.37	10.1.1.9	0				0		2.78	0	FALSE
28	10.1.1.37	10.1.1.20	0				0		2.78	0	FALSE
29	10.1.1.37	10.1.1.17	0	0	3.012812733	1.155405454	25	83.33	2.78	2.431654676	FALSE
30	10.1.1.37	10.1.1.31	0	0	0.027944	0.0055888	3	100	2.78	0.244604317	FALSE
31	10.1.1.37	10.1.1.4	0				0		2.78	0	FALSE
32	10.1.1.37	10.1.1.3	0	0	1.816564673	0.697280347	19	36.54	2.78	4.212230216	FALSE

Figure 7 – Example Output of More Features from the Dataset Produced after Stage 3

Black Hole Node Data Alteration

Examples of the training and test CSV files have been attached in the deliverable package to demonstrate the machine learning script is working. Again, it must be highlighted that many AODV characteristics have not been accurately implemented into the datasets.

It must be noted that black hole nodes were not able to be implemented in the NS-3 simulations, so the values in 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 hole node's 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 most black hole nodes are.

For example, for black hole nodes, the variable RREQs_From_Nbr is always set to 0 and its corresponding flag feature Nbr_Never_Sends_RREQ is always set to True. This is the behaviour that is expected from a black hole as it would always be responding to an RREQ with an RREP claiming to have the shortest route to the destination, so it would never broadcast the RREQ further on to its neighbours. Likewise, RREP_Resp_Pct would always be 100% because the BHN always responds to any RREQ it receives.

It would also respond with a short hop count to make it more likely to be the chosen path, so Hop_Cnt_Over_1_Cnt would always be 0. It is likely that the BHN would respond with a high destination sequence number increment to give its path the highest priority and so the flag High_Dest_Seq_Num_Inc_Pct would likely always be 100% because the BHN always responds to the RREQ with a high sequence number in the RREP.

To summarise which features have been manually altered for black hole nodes, they are Columns I to V, Y, Z & AB in the datasets.

Machine Learning from Black Hole Node Detection

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.

Only the train and test datasets have been modified. The attached examples of Stage_2 files have not been modified but they do not exhibit any black hole node behavioural characteristics.

Most of the process in the Machine learning script is explained in the actual Jupyter Notebook. So I won't repeat it in much detail here.

The training and test datasets are loaded, and there is some cleaning of the datasets. The Boolean values are converted to integers, 1 or 0 and any rows with missing values (mainly due to neighbours not carrying any traffic) are removed. It is not expected that any black hole neighbours will have rows with missing values due to the nature of their behaviour.

The features with the highest correlation to the Black_Hole_Node target variable are selected for 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.

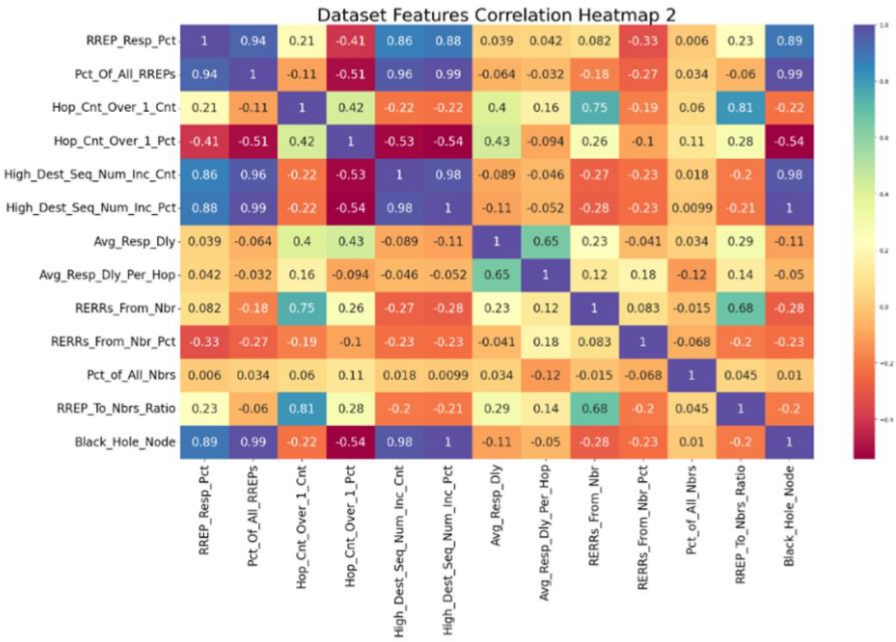


Figure 8 –Correlation heatmap demonstrating the relationship

between the features and the target variable (bottom row)

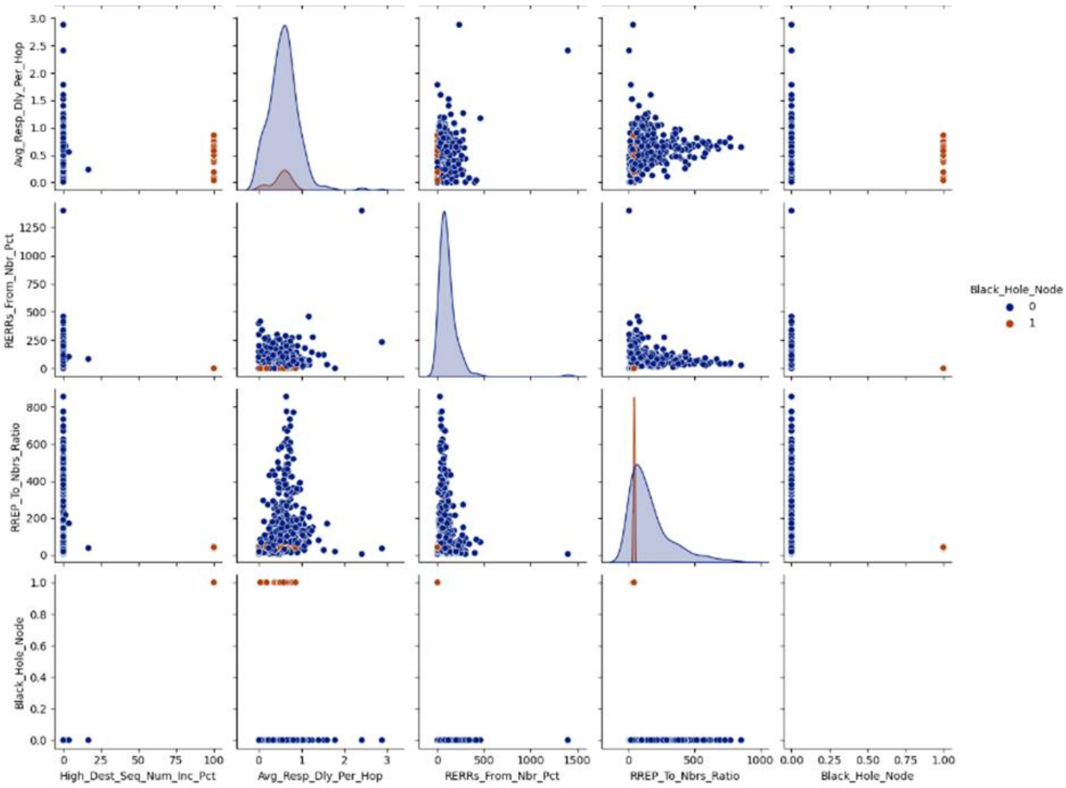


Figure 9 – Example Output of Pair Plots showing the relationships between variables.

Black hole node is brown, Normal node is Blue.

Two models are 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 is carried out on both models to determine the optimum hyperparameter settings. Standardization is also applied to the data for the SVM so that all features carry equal weight in the decision process. The models are trained and are then applied to the test set to determine their accuracy. In both models, their accuracy is 100%. This is confirmed by displaying the node IDs of the predicted rows which match those of the simulated black hole nodes.

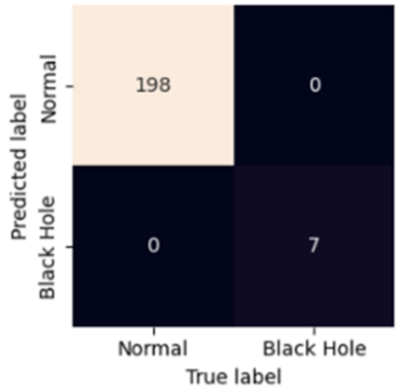


Figure 10 – Example of the Confusion Matrix from the SVM Model Classifying Black Hole Nodes.

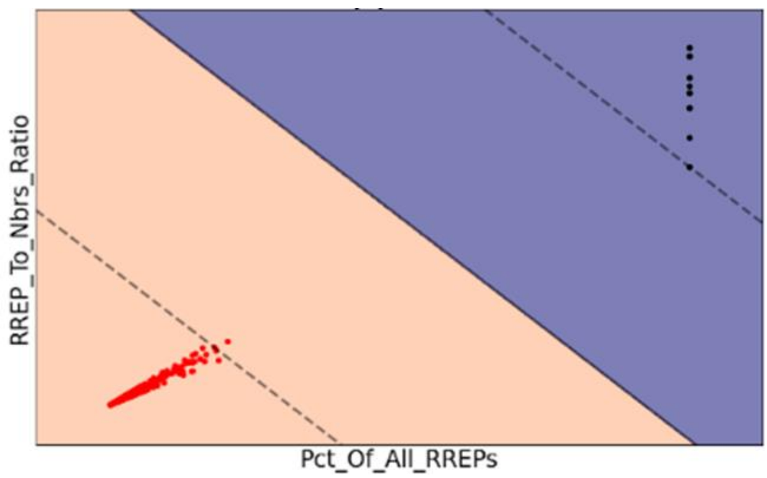


Figure 11 – Example of SVM Visualization of support vectors. There is a large separation between normal nodes (red) and black hole nodes (black).

As mentioned, more work needs to be done once the NS-3 simulator is simulating AODV networks and black hole nodes accurately. The future work section in the final report mentions what work should be carried out if anyone continues to work on this project. However, the results clearly demonstrate that machine learning can successfully be applied to AODV networks to accurately detect black hole nodes.

This script is included in the project deliverables with the name "ML_for_Malicious_Node_Detection.ipynb".

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. The time frame of the project did not permit us enough time to correct these issues.

The issues discovered with the AODV simulations are as follows:

1. The Destination Sequence Number (DSN) in the RREP message is not incremented or is not greater than the Destination Sequence Number in the corresponding RREQ. This should either be set to the destination sequence number in the RREQ + 1 or to the current sequence number in the destination node if it is already higher. This will be complex to implement as a routing table must be kept for each node which increments every time it sends an RREP.

2. The Destination Sequence Number was mostly set to 0. 0 is a special value which means the DSN is not known. This is expected at the beginning of the network simulation but the DSNs should increment afterwards.

3. The RREP does not send the Origination Sequence Number. A field needs to be created in the RREP message for this. This should be easy to implement.

4. RREPs appear to be broadcast rather than unicast. Nodes are receiving RREPs without having sent an RREQ with the same source and destination nodes.

5. Clarification needs to be made about the RREQs and RREPs from the current node. Are the messages we are seeing just what is received from the current node? Or does it include sent messages too?

6. Hellos should end in 255.255.255 not just .255. This is not a big issue.

7. Neighbour node is never the destination. Nor is the subject node. According to the destination of the RREQ.

8. All nodes have high neighbour counts 31 – 49. This is not an error, but it is a high density of nodes. Various network configurations should be made to test for black hole nodes in several network scenarios.

9. Hello Count from all neighbours should be roughly even. However, more Hellos will be generated from those nodes carrying less traffic. A HELLO message should be generated after every 1 second of inactivity (no other AODV messages sent). This latter part appears to be functioning correctly.

10. In the RREQ messages, the neighbour is never the destination. It is however observed as a destination in the RREP messages which is inconsistent.

11. Often the neighbour sees an RREQ sent from the Source Node, however, the source node's pcap file does not send it. This needs to be investigated as a high priority.

12. Similar to 11, there are inconsistencies in the messages sent in different node files. i.e. the same RREQ message is not consistent across nodes. This can be determined from its timing and Broadcast ID.

13. Black hole nodes have not been implemented yet.

It is highly likely that there are also more issues with the simulation accuracy that have not yet been identified. As the simulations are upgraded and more work continues, further testing would be carried out.

Future Work

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, the black hole nodes should be "dumb" where their behaviour is obvious, i.e., they act exactly as described above.
- More features could be extracted from the trace files and added to the dataset that go beyond the AODV protocol which includes the amount of traffic data (kilobytes, megabytes) received from 1st tier neighbour node.
- 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
    - The number of black hole nodes.
- Combine all of these datasets of the various network configurations into one large training dataset. This larger set will be used to train the models to various configurations of networks, so that it can 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, the aim is to deploy these algorithms into the normal nodes within the network. These normal nodes will monitor their first-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.