

Assignment 1

Course: AI800 - Research Methodology in Computer Science

Student Name: Naif Ahmed Basalib

Date: March 2026

Table of Contents

1. Introduction	3
2. Review of Selected Papers	4
2.1 Gao et al. (2019)	4
2.2 Poongodi et al. (2019)	6
2.3 Polat et al. (2021)	8
2.4 Al-Mehdhara and Ruan (2021)	10
2.5 Turkoğlu et al. (2022)	12
2.6 Kadam and Sekhar (2021)	14
3. Connections and Conflicts	16
4. Summary Table	18
5. Synthesis	19
6. References	21

1. Introduction

Vehicular Ad Hoc Networks, or VANETs, are a type of wireless network where vehicles communicate with each other and with roadside units to share information about traffic, road conditions, and emergencies. The main goal of VANETs is to improve road safety and make transportation more efficient. In recent years, VANETs have become an important part of what researchers call the Intelligent Transportation System (ITS). However, because they use wireless communication and have an open network structure, they are vulnerable to many types of cyber attacks.

One of the most dangerous attacks on VANETs is the Distributed Denial of Service (DDoS) attack. In a DDoS attack, multiple compromised nodes in the network send a huge amount of traffic to a target node, making it unable to respond to legitimate requests. In a VANET context, this could stop important safety messages from being delivered, which can be very dangerous. Because of this, detecting and stopping DDoS attacks in VANETs is a very active area of research.

One reason this problem is hard to solve is that VANETs have some unique characteristics. Vehicles move fast, so the network topology changes all the time. A vehicle might only stay in range of another vehicle for a few seconds. This makes it difficult to use traditional intrusion detection systems that were designed for fixed networks. Researchers have tried many different approaches, including machine learning, deep learning, trust-based systems, and Software-Defined Networking (SDN).

This assignment reviews six papers that propose different methods for detecting DDoS attacks in VANETs. The papers I reviewed were published between 2019 and 2022, so they give a fairly recent view of how this topic has been studied. For each paper, I will provide a summary of the main idea and method, followed by a critical analysis of the strengths and weaknesses. After that, I will discuss how the papers relate to each other, present a summary table, and end with a synthesis of what I learned from reading all six papers.

2. Review of Selected Papers

2.1 Gao et al. (2019) - A Distributed Network Intrusion Detection System for DDoS Attacks in VANET

2.1.1 Summary

This paper was published in IEEE Access in 2019. The authors argue that the main problem with existing intrusion detection systems for VANETs is that they are centralised, meaning all the detection work happens in one place. This becomes a bottleneck when the network is large, because the single detection node cannot process all the traffic fast enough. To solve this, Gao et al. propose a distributed intrusion detection system that uses big data technology.

The proposed system is built using Apache Hadoop and Apache Spark. Hadoop is used to store traffic data in a distributed way across many nodes using its file system called HDFS. Spark is then used to process the data in parallel, which makes it much faster than processing it on a single machine. The actual detection is done by a Random Forest classifier, which works by building many decision trees and combining their results.

The system was tested on two datasets: NSL-KDD and UNSW-NB15. The results were very good. On NSL-KDD, the accuracy was 99.95%, and on UNSW-NB15, it was 98.75%. The authors also compared their distributed system with a centralised one and showed that their approach is faster. They concluded that the combination of big data processing and Random Forest is a good solution for detecting DDoS attacks in VANETs.

2.1.2 Critical Analysis

The main strength of this paper is the idea of making the detection distributed. This makes sense for VANETs because a centralised system would create a single point of failure. The use of Random Forest is also a good choice because it handles noisy data well and is hard to fool with small changes in the input.

That said, I still noticed a few weaknesses in the paper. To me, the biggest problem is that neither dataset is actually specific to VANETs. NSL-KDD is a very old dataset that comes from a completely different type of network. It does not represent the kind of traffic you would find in a vehicular network, where vehicles are moving and connections are short. Using such a dataset makes it hard to know if the system would actually work in a real VANET.

Another problem is that deploying Hadoop and Spark requires a lot of computing resources. Vehicles have limited processing power and battery life, so running a big data framework on them may not be practical. The authors do not discuss this issue at all. Also, the paper only reports accuracy and does not give information about false positives or false negatives, which are important metrics for a security system.

2.2 Poongodi et al. (2019) - DDoS Detection Using a Trust-Based Evaluation System in VANET

2.2.1 Summary

This paper takes a very different approach compared to Gao et al. Instead of using machine learning to classify traffic, Poongodi et al. use a trust-based system. The idea is that each node in the network is given a trust score based on how it behaves. If a node starts acting suspiciously, for example by sending too many packets, its trust score goes down. If the trust score falls below a certain level, the node is considered malicious and is removed from the network.

To manage the network, the authors use a clustering technique called DGSOT, which stands for Distributed Genetic Spiral Optimisation Technique. This algorithm groups nearby vehicles into clusters and selects a cluster head for each group. The cluster head is responsible for monitoring the behaviour of the other nodes in its cluster and updating their trust scores.

The system was evaluated using NS-2 simulation. The results are reported in terms of Packet Delivery Ratio (PDR), end-to-end delay, and energy consumption. The paper shows that their system performs better than a network without any protection in all three metrics, meaning more packets are delivered, with lower delay and less energy used.

2.2.2 Critical Analysis

One advantage of the trust-based approach is that it does not need labelled training data. Unlike machine learning methods that need examples of attack traffic to learn from, this system just watches how nodes behave and flags anything that looks unusual. This could make it useful against new types of attacks that the system has never seen before.

I think the evaluation is less convincing than the others because the paper does not report standard metrics like accuracy, precision, or recall. The authors do not report detection accuracy, precision, or recall, which makes it hard to compare this system with others. Just knowing that PDR improved does not tell us how many attackers were actually detected or how many legitimate nodes were wrongly flagged as malicious.

I also think the cluster head design creates a security risk. If an attacker is able to become the cluster head, they could manipulate the trust scores and prevent real attackers from being detected. The paper does not address this scenario. Additionally, running a genetic-based optimisation algorithm on every cluster head could be computationally expensive for resource-limited vehicles.

2.3 Polat et al. (2021) - Deep Network Approach with Stacked Sparse Autoencoders in SDN-Based VANET

2.3.1 Summary

Polat et al. propose a deep learning method for detecting DDoS attacks in a Software-Defined VANET. The main contribution of this paper is the use of Stacked Sparse Autoencoders (SSAE) combined with a Softmax classifier. An autoencoder is a type of neural network that learns to compress data into a smaller representation and then reconstruct it. The 'sparse' part means that only a small number of neurons are active at any time, which helps the model learn more useful features. 'Stacked' means multiple autoencoders are connected in layers.

The system is designed for an SDN-based VANET, where a centralised SDN controller manages the whole network. This gives the detection system a full view of all the traffic in the network, which makes detection easier. The SSAE is first trained to learn what normal traffic looks like. Then a Softmax classifier is added on top to separate normal traffic from attack traffic.

The authors created their own simulation dataset for evaluation. The reported accuracy was 96.9%. They compared their method against some traditional machine learning methods and showed that the deep learning approach performed better. The paper argues that automatically learning features from data is better than manually selecting features.

2.3.2 Critical Analysis

Using deep learning for this problem is a good idea because deep neural networks can automatically extract useful features from raw data without needing much manual feature engineering. The autoencoder is especially suitable here because it is good at learning the structure of normal traffic and then detecting things that do not fit. This is a strength of the method.

However, the SDN-based architecture is a concern. VANETs are supposed to be decentralised networks where vehicles communicate directly. Relying on a centralised SDN controller means that if the controller goes down or loses connection, the detection system stops working. In a highly mobile environment like a VANET, this is a real problem. The paper does not explain what happens in this situation.

Another issue is that the authors used their own custom dataset instead of a publicly available one. This makes it very hard to compare their results with other papers. We do not know enough details about how the dataset was created, how many samples it has, or what types of attack traffic it includes. The 96.9% accuracy looks good, but it is difficult to interpret without more information about the dataset.

2.4 Al-Mehdhara and Ruan (2021) - Multilayer Distributed SOM for DDoS Detection in SDN-Based VANET

2.4.1 Summary

Al-Mehdhara and Ruan propose a system called MSOM, which stands for Multilayer Distributed Self-Organising Map. A Self-Organising Map (SOM) is a type of unsupervised neural network that organises data into a grid based on similarity. The 'multilayer' part means that multiple SOM layers are stacked, allowing the system to learn more complex patterns. The 'distributed' part means that instead of having one central SOM, there are multiple local SOM instances placed throughout the network.

Like Polat et al., this system is also designed for an SDN-based VANET. The SDN controller collects traffic data from all parts of the network and feeds it into the MSOM. The SOM is first trained to learn the patterns of normal traffic. During detection, traffic that does not match the learned normal patterns is flagged as an attack. Because the system is unsupervised, it does not need labelled attack data for training.

The system was evaluated using NS-3 simulation. The reported detection accuracy was 99.67%, which is one of the highest figures among all the papers reviewed here. The authors also report that their system has good performance in terms of detection speed and network overhead.

2.4.2 Critical Analysis

The use of Self-Organising Maps for anomaly detection is interesting. SOMs are good at finding patterns in unlabelled data, which means this system could potentially detect new types of DDoS attacks that were not seen during training. The distributed design also helps to reduce the load on the central controller, which is an improvement over some other SDN-based approaches.

However, just like the previous paper, the dependence on SDN is a limitation. If there is a problem with the SDN controller, the whole detection system could fail. The authors discuss the distributed nature of the SOM in some detail, but they still rely on the controller to collect data and coordinate the system.

One more concern is that SOMs can have problems when the data is very unbalanced. In a real VANET, DDoS attacks are hopefully rare, so most of the data during training will be normal traffic. If the system only sees very few attack examples, it might not learn to recognise attacks well. The paper does not discuss this issue. Also, the use of NS-3 simulation, while better than simple models, is still not the same as a real-world deployment.

2.5 Turkoğlu et al. (2022) - MRMR Feature Selection and Bayesian Optimisation for DDoS Detection in SD-VANET

2.5.1 Summary

This is the most recent paper in this review. Turkoğlu et al. work with a dataset from a Software-Defined VANET that contains 17,779 labelled samples. What makes this paper different from the others is that it focuses a lot on how to choose the right features and how to tune the machine learning models properly before training them.

For feature selection, they use a method called MRMR, which stands for Minimum Redundancy Maximum Relevance. This method selects features that have a strong relationship with the class label (whether traffic is normal or an attack) while avoiding features that are too similar to each other. The idea is that having too many redundant features can slow down training and reduce accuracy.

For hyperparameter tuning, the authors use Bayesian optimisation, which is a smart way of finding the best settings for a machine learning model. Instead of trying every possible combination of settings (which would take too long), Bayesian optimisation uses a probabilistic model to decide which settings to try next. They test three classifiers: K-Nearest Neighbour (KNN), Support Vector Machine (SVM), and Decision Tree (DT). The best result was 99.35% accuracy with SVM.

2.5.2 Critical Analysis

I think this paper is one of the more careful and honest papers in the review. Many papers in this area just pick a classifier and report the best accuracy they can find without really explaining their choices. This paper actually explains why they chose MRMR for feature selection and why Bayesian optimisation is better than grid search. This makes the results more trustworthy.

Using an SD-VANET specific dataset is also a good point. Unlike papers that use old general-purpose datasets like NSL-KDD, this dataset was created in a vehicular network context. This means the traffic features are more representative of what would happen in a real VANET. The dataset size of 17,779 samples is also reasonable.

I felt the comparison here was a bit narrow because the authors only tested KNN, SVM, and Decision Tree. Since deep learning is already common in this area, including at least one neural model would have made the comparison stronger. I think the paper would have been stronger if it had included at least one deep learning model, because that would make the comparison with classical machine learning more convincing. Also, even though the accuracy is high at 99.35%, the paper does not discuss the false negative rate, which tells us how many attacks were missed. In a security system, missing an attack is a serious problem, so this metric should be reported.

2.6 Kadam and Sekhar (2021) - Hybrid KSVM Algorithm for DDoS Detection in VANET

2.6.1 Summary

Kadam and Sekhar propose a hybrid machine learning method called KSVM. The idea is to combine two popular algorithms: K-Nearest Neighbour (KNN) and Support Vector Machine (SVM). The authors argue that SVM alone can be slow when the dataset is very large, and KNN alone can be inaccurate if the neighbourhood is not well-defined. By combining them, they hope to get the advantages of both methods.

The way the KSVM works is that SVM is first used to find the most important training samples, called support vectors. Then KNN uses only these support vectors to classify new traffic samples instead of searching through the entire dataset. This makes the process faster and hopefully more accurate.

They collected data from Kaggle, a popular data science platform. The dataset contains DDoS traffic samples with features like source and destination IP address, port numbers, and communication protocol. They trained the model on 70% of the data and tested on the remaining 30%. The KSVM achieved 92.46% accuracy, which was better than standalone SVM (92%), KNN (86%), Decision Tree (89%), and ANN (90.23%). Precision was 95.31% and recall was 95.67%.

2.6.2 Critical Analysis

The hybrid approach is a reasonable idea and the paper does a good job of comparing it against other individual classifiers. Reporting precision and recall alongside accuracy is also better than just reporting accuracy, because it gives a more complete picture of the model's performance.

I was not fully convinced by this result because the gain over the standalone SVM model is very small. Going from 92% to 92.46% is only about 0.46%, which may not be worth the added complexity of maintaining two models instead of one. To me, the paper does not clearly show why this small improvement would make a real difference in practice.

What stood out more to me was the use of a Kaggle dataset rather than one built specifically for VANET traffic. General DDoS traffic from the internet is very different from DDoS traffic in a vehicular network. VANETs use different protocols (like DSRC/WAVE), have different traffic patterns, and vehicles move around constantly. A model trained on general internet traffic may not work well when deployed in a real VANET. This is a significant limitation that the authors do not really address.

3. Connections and Conflicts Among the Papers

After reading all six papers, I noticed some common themes and also some important differences. In this section, I will discuss what the papers agree on and where they disagree.

3.1 Common Themes

All six papers agree that DDoS attacks are a serious threat to VANETs and that something needs to be done to detect and stop them. They also all agree that the unique characteristics of VANETs, such as high mobility and changing network topology, make it hard to use traditional security methods. This is the motivation behind all six papers.

Another common point is that machine learning is useful for this problem. Five out of six papers use some form of machine learning, including Random Forest, deep learning, SOM, SVM, and hybrid classifiers. Even the trust-based paper by Poongodi et al. uses pattern analysis, which is related to machine learning thinking. This shows that the research community has largely moved away from rule-based detection and toward data-driven approaches.

3.2 Disagreements on Network Architecture

One of the biggest differences between the papers is the choice of network architecture. Three papers (Polat et al., Al-Mehdhar and Ruan, and Turkoğlu et al.) design their systems for SDN-based VANETs, where a central controller manages the network. The other three papers (Gao et al., Poongodi et al., and Kadam and Sekhar) work with a normal distributed VANET without a central controller.

SDN-based approaches can detect attacks more easily because the controller has a view of the whole network. But this creates a dependency on the controller that may not be realistic in a real VANET where vehicles are constantly moving. If the vehicle moves out of range of the controller, the detection system may stop working. The distributed papers avoid this problem but can only look at traffic from a single node's perspective.

3.3 Different Datasets Make Comparison Difficult

A big problem with comparing these six papers is that each one uses a different dataset. Gao et al. use NSL-KDD and UNSW-NB15. Kadam and Sekhar use a Kaggle dataset. Polat et al. and Al-Mehdhar and Ruan use custom simulation datasets. Turkoğlu et al. use an SD-VANET dataset. And Poongodi et al. do not use a separate dataset at all, they just report simulation metrics.

Because the datasets are all different, we cannot really compare the accuracy numbers from one paper to another. A system that gets 99.95% on NSL-KDD and a system that gets 96.9% on a custom VANET dataset may both be good systems, but we cannot say which one is better just from the numbers. This is a general problem in this research area and it would be very helpful if researchers agreed on a standard benchmark dataset.

3.4 High Accuracy in Simulation vs. Reality

Most papers in this review report very high accuracy, often above 99%. However, these results are all from simulation environments or controlled datasets. In a real VANET with real vehicles and real network conditions, there would be much more noise and uncertainty. Papers that report lower accuracy like Kadam and Sekhar (92.46%) but discuss their limitations more honestly may actually be more realistic about what is achievable in practice.

4. Summary Table of Reviewed Papers

Author & Year	Method	Architecture	Dataset	Best Accuracy	Simulation
Gao et al. (2019)	Random Forest + Hadoop/Spark	Distributed VANET	NSL-KDD, UNSW-NB15	99.95%	Not stated
Poongodi et al. (2019)	Trust-based + DGSOT	Distributed VANET	NS-2 Simulation	N/A	NS-2
Polat et al. (2021)	Stacked Sparse Autoencoders + Softmax	SDN-VANET	Custom SDN-VANET dataset	96.9%	Custom
Al-Mehdhara & Ruan (2021)	Multilayer SOM (MSOM)	SDN-VANET	NS-3 Simulation	99.67%	NS-3
Turkoğlu et al. (2022)	MRMR + Bayesian Opt. + SVM	SDN-VANET	SD-VANET (17,779 samples)	99.35%	N/A
Kadam & Sekhar (2021)	Hybrid KSVM (KNN + SVM)	Distributed VANET	Kaggle DDoS	92.46%	Python

5. Synthesis

After reviewing all six papers, I can see that there has been significant progress in detecting DDoS attacks in VANETs, but many challenges still remain. In this section, I will summarise the most important findings and suggest directions for future research.

One thing that stood out to me was how heavily these papers rely on machine learning as the main way to tackle the problem. Most papers in this review use some form of ML, ranging from traditional classifiers like Random Forest and SVM to deep learning methods like autoencoders. The results show that ML can achieve very high accuracy in simulation environments. However, it is not yet clear how well these systems would work in real-world deployments, since all the evaluations were done either in network simulators or using datasets that may not represent real VANET traffic.

Another thing I noticed is that the network architecture makes a big difference in how these systems are designed and evaluated. I noticed that the SDN-based papers usually report better accuracy, probably because the centralized controller can see the network more clearly as a whole. I think that becomes much harder to apply in a highly mobile environment like VANETs. I found the distributed approaches more realistic for actual VANETs, but they also have to make decisions with much less information available. I think a more practical direction for future work would be hybrid architectures, where nodes share some useful information without depending on a fully centralized controller.

Third, it became clear to me that this research area still lacks a standard benchmark dataset. What makes comparison difficult is that every paper relies on a different dataset, so the results are not really measured on equal ground. For example, Gao et al. report 99.95% accuracy on NSL-KDD, but I do not think that automatically makes their system the strongest, since NSL-KDD is an old dataset and not designed specifically for VANET traffic. I think progress in this area would be much stronger if researchers had one public VANET DDoS dataset they could all test against.

Fourth, these papers pay much more attention to accuracy than to other metrics that also matter in practice. In my view, accuracy alone is not enough here, because we also need to know how many attacks the system misses and how often it wrongly flags normal traffic. In a VANET, missing even one real attack could interrupt an important safety message, and that can have serious consequences. I think future studies would be much more useful if they reported these metrics more clearly and in more detail.

Finally, what stood out to me is that none of these papers seriously discuss adversarial robustness. I think a realistic attacker would probably try to make the malicious traffic look normal once they know a detection system is being used. In other words, the attacker may try to fool the detection system itself, not just attack the network directly. I think future research should look more seriously at how these systems hold

up when attackers actively try to avoid detection. Overall, I came away feeling that this area has made progress, but it is still not fully ready for real-world VANET deployment.

6. References

- [1] Y. Gao, H. Feng, and Z. Li, "A Distributed Network Intrusion Detection System for Distributed Denial of Service Attacks in Vehicular Ad Hoc Network," *IEEE Access*, vol. 7, pp. 154560-154571, 2019.
- [2] M. Poongodi, A. Sharma, V. Vijayakumar, V. Bhardwaj, P. Sharma, R. Iqbal, and R. Kumar, "DDoS Detection Mechanism Using Trust-Based Evaluation System in VANET," *IEEE Access*, vol. 7, pp. 87223-87240, 2019.
- [3] H. Polat, O. Polat, and A. Cetin, "Detecting DDoS Attacks in Software-Defined Networks Through Feature Selection Methods and Machine Learning Models," *IET Communications*, vol. 15, no. 4, pp. 521-535, 2021.
- [4] H. Al-Mehdhara and L. Ruan, "A Multilayer Distributed Self-Organising Map for DDoS Attack Detection in Software Defined Vehicular Networks," *Security and Communication Networks*, Hindawi, vol. 2021, Article ID 8891758, 2021.
- [5] M. Turkoğlu, C. Birant, and D. Birant, "DDoS Attacks Detection in Software-Defined Vehicular Networks Using Machine Learning," *Expert Systems with Applications*, vol. 204, p. 117483, 2022.
- [6] N. Kadam and K. R. Sekhar, "Machine Learning Approach of Hybrid KSVN Algorithm to Detect DDoS Attack in VANET," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 12, no. 7, pp. 718-722, 2021.