|  |  |  |
| --- | --- | --- |
|  | **BLOCKCHAIN-BASED WIRELESS SENSOR**  **NETWORKS FOR MALICIOUS NODE DETECTION** |  |

# PHASE I PROJECT REPORT

***Submitted by***

|  |  |
| --- | --- |
| **SIVASUNDARESAN S** | **(20110098)** |
| **SRI SANTHOSH N** | **(20110104)** |
| **SURESH K** | **(20110114)** |
| **SARAVANARAJA G** | **(20110804)** |

***in partial fulfillment for the award of the degree of***

**BACHELOR OF TECHNOLOGY**

***in***

# INFORMATION TECHNOLOGY

**HINDUSTHAN COLLEGE OF ENGINEERING AND TECHNOLOGY**

Approved by AICTE, New Delhi, Accredited with ‘A’ Grade by NAAC **(An Autonomous Institution, Affiliated to Anna University, Chennai)** Valley Campus, Pollachi Highway, Coimbatore – 641 032

# DECEMBER 2023

***Hindusthan College of Engineering and Technology***

Approved by AICTE, New Delhi, Accredited with ‘A’ Grade by NAAC **(An Autonomous Institution, Affiliated to Anna University, Chennai)** Valley Campus, Pollachi Highway, Coimbatore – 641 032

**BONAFIDE CERTIFICATE**

Certified that this project report **“BLOCKCHAIN-BASED WIRELESS SENSOR NETWORKS FOR MALICIOUS NODE DETECTION:A SURVEY”** is the bonafide work of **SIVASUNDARESAN S (20110098), SRI SANTHOSH N (20110104), SURESH K (20110114), SARAVANARAJA G (20110804)** who

carried out the project work under my supervision.

|  |  |
| --- | --- |
| **SIGNATURE** | **SIGNATURE** |
| **SABARIGIRIRAJ**  **HEAD OF THE DEPARTMENT**  Information Technology Hindusthan College of Engineering and Technology, Coimbatore-32 | **Ms. M.MASILAMANI SUPERVISOR**  **Assistant Professor** Information Technology Hindusthan College of Engineering and Technology,  Coimbatore-32 |



Submitted for the Autonomous Institution Project Phase-1 Viva-Voce conducted on

# INTERNAL EXAMINER EXTERNAL EXAMINER

DECLARATION

We, hereby jointly declare that the project work entitled **“BLOCKCHAIN- BASED WIRELESS SENSOR NETWORKS FOR MALICIOUS NODE**

**DETECTION:A SURVEY”,** submitted to the Autonomous Institution Project Phase I Viva Voice - December 2023 in partial fulfillment for the award of the degree of **BACHELOR OF TECHNOLOGY IN INFORMATION**

**TECHNOLOGY**, is the report of the original project work done by us under the guidance of **Ms.M.MASILAMANI ,** Assistant Professor, Department of Information Technology, Hindusthan College of Engineering and Technology, Coimbatore.

**NAME SIGNATURE**

**SIVASUNDARESAN S**

**SRI SANTHOSH S**

**SURESH K**

**SARAVANARAJA G**

I certify that the declaration made by the above candidates are true.

Project Guide,

Ms. M. Masilamani, Assistant Professor, Department of IT,

Hindusthan College of Engineering and Technology, Coimbatore –32

ACKNOWLEDGEMENT

We take this opportunity to express our whole hearted thanks and our profound respect to all those who guided and inspired us in the completion of this phase I project work.

We extend our sincere thanks to the Founder and Chairman of Hindusthan Educational and Charitable Trust **Shri. T. S. R. Khannaiyann** and the Managing Trustee **Smt. Sarasuwathi Khannaiyann** and Executive Trustee & Secretary **Mrs. Priya Satish Prabhu** for providing essential infrastructure.

We would like to reveal our profound thanks to our respected Principal,**Dr.**

**J. Jaya,** who happens to be striving force in all endeavours.

We would like to express our gratitude to the Head of the Department **Sabarigiriraj.,** for bringing out the project successfully and for strengthening the ray of hope.

We would like to express our sincere thanks and deep sense of gratitude to our Guide **Ms. M. Masilamani,** Assistant Professor, Information Technology, for her valuable guidance, suggestions and constant encouragement which paved way for the successful completion of the project work.

We express our immense pleasure and thankfulness to our Class Advisor and Project Coordinator, **Ms. M. Masilamani,** Assistant Professor, and all other Faculty members of the Department of Information Technology, technical staffs and friends who helped us for the successful completion of this project. We express our earnest gratitude to our family members who encouraged us and strengthened us in perilous path, encountered during our task.

# ABSTACT

This project presents a framework for converting wireless signals into structured datasets, which can be fed into machine learning algorithms for the detection of active eavesdropping attacks at the physical layer. More specifically, a wireless communication system, which consists of an access point (AP), K legitimate users and an active eavesdropper, is considered. To detect the eavesdropper who breaks into the system during the authentication phase, we first build structured datasets based on different features and then apply sophisticated support vector machine (SVM) classifiers to those structured datasets.

To be more specific, we first process the signals received by the AP and then define a pair of statistical features based on the post-processing of the signals. By arranging for the AP to simulate the entire process of transmission and the process of constructing features, we form the so-called artificial training data (ATD).

Recent advances in the digital actual shrewd framework (CPSG) have empowered a wide scope of new gadgets dependent on the data and correspondence innovation (ICT). Nonetheless, these ICT-empowered gadgets are powerless to a developing danger of digital actual assaults. This paper plays out an intensive audit of the best in class digital actual security of the savvy framework. By zeroing in on the actual layer of the CPSG, this paper gives a preoccupied and bound together state-space model, in which digital actual assault and protection models can be successfully summed up. The current digital actual assaults are sorted as far as their target parts.

We then, at that point talk about a few functional and enlightening protection moves toward that present the present status of-the-workmanship in the field, including moving objective guard, watermarking, and information driven approaches. At long last, we examine difficulties and future freedoms related with the shrewd matrix cyber physical security.

(CPSs) are smart systems that include engineered interacting networks of physical and computational components. The comprehensively interconnected and integrated systems contribute new functionalities to enable technological development in critical infrastructures, such as electric power systems, water networks, transportation, home automation, and health care. A CPS encompasses complex systems of control, awareness, computing, and communication. The complexity and heterogeneity have indicated the potential challenges to the security and resilience of CPSs. The interconnection of bulk physical layer components is challenging the protection against inherent physical vulnerabilities therein. On the other hand, cyber-integration, which relies on network communication and the internet of things (IoT) based devices, requires extraordinary investments in security designs and upgrades against unanticipated threats from cyberspace. A cyber-physical attack is defined as a security breach in cyberspace that adversely affects the physical space of a CPS. Cyber-physical attacks compromise the confidentiality, integrity, and availability of information by coupling cyber and physical spaces in a CPS. In the past decades, several noteworthy cyber-physical attacks have been reported in the industry, facilitating synergistic efforts from industry practitioners and research communities towards a new CPS security era. The first proclaimed cyber-physical attack dated back to 1982 in the Siberian wilderness, where attackers manipulated the pipeline control software, which led the valves’ control to misbehave, resulting in severe crossing of pressure limits and eventually a massive explosion .

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO** | **TITLE** | **PAGE NO** |
|  | **ABSTRACT** | **i** |
|  | **List of Figures** | **iv** |
|  | **List of Tables** | **v** |
| **1** | **INTRODUCTION** | **1** |
|  | 1.1 Project Overview | **1** |
|  | 1.2 Purpose of Project | **2** |
| **2** | **LITERATURE SURVEY** | **8** |
|  | 2.1 Overview | **3** |
|  | 2.2 Existing Problem | **6** |
|  | 2.3 Problem Statement Definition | **7** |
| **3** | **IDEATION & PROPOSED SOLUTION** | **8** |
|  | 3.1 Ideation | **8** |
|  | 3.2 Empathy Map Canvas | **9** |
|  | 3.3 Ideation & Brainstorming | **10** |
|  | 3.4 Proposed Solution | **11** |
|  | 3.5 Problem Solution Fit | **13** |
| **4** | **REQUIREMENT ANALYSIS** | **14** |
|  | 4.1 Functional Requirement | **14** |
|  | 4.2 Non Functional Requirements | **15** |
| **5** | **PROJECT DESIGN** | **16** |
|  | 5.1 Data Flow diagrams | **16** |

**INTRODUCTION**

Wireless sensor networks (WSNs) are generally composed of dispersed micro-devices (termed sensors), which may be embedded and possess simple or various sensing capabilities. These networks are widely used in various areas such as smart homes, military and industrial applications due to their wide range of coverage areas support, massive precision monitoring, remote monitoring, fast stabilization, high fault tolerance and ease of use and unique characteristics including self-organization.

However, sensor nodes are inherently limited, as they can only permit limited energy, processing capability, transmission range, and memory on board. Moreover, as a consequence of these limitations, sensor nodes are vulnerable to compromise. Risks facing WSN security often arise from outside and inside the network, in which the proper network nodes are compromised and sometimes forced to act as malicious nodes.

The ability to detect, contain, and purge in-network malicious nodes in good time is an equally essential concern for WSN security. Resolving the issues related to security has had a profound impact on the design and development trends of WSNs and has attracted wide attention in the literature. Furthermore, various mechanisms for malicious nodes detection in wireless sensor networks have been proposed. For instance, Min and Ranxin introduced a technique to detect Malicious node Detection using a Triangle Module fusion Operator (MDTMO), which can check selective forward attacks.

This common network attack makes nodes discard all or certain data packets selectively so that the cluster head and base station cannot receive the full monitoring data. The MDTMO technique establishes packet loss sendreceive and receive-forward membership functions based on the information packets sent to a node, obtained by the node, and forwarded by the same node. It then uses the

triangle module fusion technique on the membership functions to detect a potentially malicious device.

The base station (BS) device is alerted of the potentially compromised node, detects its channel and buffer occupancy, and tries to assess if the packet drops are due to network congestion. Suppose the WSN quality is better than the result in such congestion and data loss. In that case, it is regarded that the packets loss comes from a selective forward attack rather than jamming, and the device is labelled malicious.

# BLOCKCHAIN TECHNOLOGY

Blockchain technology is a revolutionary and decentralized system that has redefined the way data is stored, shared, and secured in the digital age. At its core, blockchain is a distributed ledger that records transactions across a network of computers in a transparent and tamper-resistant manner. Unlike traditional centralized systems, where a single entity holds control, blockchain operates on a peer-to-peer network, providing a transparent and trustless environment.

The foundation of blockchain lies in its blocks, which are containers for data, and the chain, which links these blocks together in a chronological order. Each block contains a cryptographic hash of the previous block, creating a chain that is virtually impossible to alter retroactively. This immutability ensures the integrity of the data stored within the blockchain.

One of the key features of blockchain is its consensus mechanism, a protocol that enables participants in the network to agree on the state of the ledger. The most well-known consensus mechanism is Proof of Work (PoW), used by cryptocurrencies like Bitcoin, where participants (miners) compete to solve complex mathematical problems to validate transactions and add blocks to the chain. Other consensus mechanisms, such as Proof of Stake (PoS) and Delegated Proof of Stake (DPoS), have emerged as more energy-efficient alternatives.

Blockchain technology's applications extend beyond cryptocurrencies, encompassing a wide array of industries. Smart contracts, self-executing contracts with the terms of the agreement directly written into code, automate and enforce contractual agreements, reducing the need for intermediaries. Industries like finance, supply chain, healthcare, and more are exploring the potential of blockchain to enhance transparency, traceability, and security in their operations.

In conclusion, blockchain technology represents a paradigm shift in the way information is managed and transactions are conducted. Its decentralized nature, immutability, and transparency make it a powerful tool with far-reaching implications for various sectors, promising increased efficiency, security, and trust in the digital landscape.

# MALICIOUS NODE DETECTION

Malicious node detection is a critical aspect of securing networks and ensuring the integrity of communication in various distributed systems. As technology continues to advance, the prevalence of malicious activities within networks has become a significant concern. Malicious nodes, which are nodes in a network that exhibit harmful behavior, pose a threat to the proper functioning of communication protocols, data integrity, and overall network security.

The increasing complexity and interconnectedness of modern networks make them susceptible to a variety of attacks, ranging from simple packet sniffing to more sophisticated denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks. Identifying and isolating these malicious nodes is crucial for maintaining the reliability and trustworthiness of networked systems.

The detection of malicious nodes involves the development and implementation of algorithms, techniques, and strategies to differentiate between normal and malicious behavior within a network. This field has gained significance in the context of wireless sensor networks, ad-hoc networks, the

Internet of Things (IoT), and other distributed computing environments where the potential impact of malicious nodes is particularly pronounced.

In this introduction, we will explore the challenges associated with detecting malicious nodes, the importance of timely detection, and the role of advanced technologies such as machine learning and anomaly detection in addressing these challenges. As the landscape of cyber threats continues to evolve, a robust malicious node detection system becomes paramount for safeguarding the integrity and functionality of networked systems. This introduction sets the stage for a deeper exploration into the methodologies and technologies employed in the ongoing battle against malicious nodes in distributed networks.

# NETWORK SECURITY MANAGEMENT

Network security management is a critical aspect of modern information technology, playing a pivotal role in safeguarding the integrity, confidentiality, and availability of data within interconnected systems. As organizations increasingly rely on complex networks to conduct their operations, the need for robust security measures becomes paramount. The dynamic and interconnected nature of modern networks introduces a plethora of vulnerabilities that can be exploited by malicious actors, making it imperative for businesses to adopt comprehensive network security strategies.

The goal of network security management is to create a secure and resilient environment that protects sensitive information, prevents unauthorized access, and ensures the continuous functionality of critical systems. This involves implementing a multi-faceted approach that includes a combination of hardware, software, policies, and practices to detect, prevent, and respond to potential threats. In essence, network security management encompasses a proactive stance against cyber threats, acknowledging the ever-evolving landscape of risks and adapting to emerging challenges.

In this context, it is essential to recognize the interconnectedness of devices and systems within a network, ranging from traditional computers and servers to mobile devices, Internet of Things (IoT) devices, and cloud-based services. The complexity introduced by this diverse ecosystem requires a holistic understanding of potential vulnerabilities and the implementation of strategic measures to mitigate risks. Network security management is not a one-size-fits-all solution but a tailored and adaptive process that considers the specific needs and risks faced by an organization.

This introduction sets the stage for a deeper exploration of network security management, delving into the key components, strategies, and technologies that contribute to a robust and resilient security posture. As technology continues to advance, the importance of effective network security management will only grow, making it an indispensable aspect of any organization's overall risk management and information protection strategy.

# DISTRIBUTED CONSENSUS ALGORITHM

Distributed Consensus Algorithms play a crucial role in ensuring the reliability and consistency of distributed systems. As the demand for scalable and fault-tolerant systems has increased, the need for effective consensus mechanisms has become paramount. In a distributed system, where multiple nodes collaborate to achieve a common goal, consensus ensures that all nodes agree on a single, consistent state despite potential failures and network delays.

The fundamental challenge in distributed systems is achieving consensus among nodes that may have differing information due to failures or delays. This challenge becomes particularly pronounced in scenarios where nodes need to agree on a decision, such as reaching a consistent state, electing a leader, or committing to a transaction. Distributed Consensus Algorithms address these challenges by providing a systematic approach for nodes to reach agreement in the presence of faults and uncertainties.

One of the earliest and most well-known problems in distributed consensus is the Byzantine Generals Problem, which explores how to achieve consensus among a group of nodes that may include malicious or faulty actors. The development of practical consensus algorithms has evolved to address this problem and ensure that distributed systems can operate reliably in real-world scenarios.

Prominent examples of distributed consensus algorithms include Paxos, Raft, and Practical Byzantine Fault Tolerance (PBFT). These algorithms provide different approaches to solving the consensus problem, each with its strengths and weaknesses. Paxos, for instance, focuses on a simple and provably correct protocol, while Raft emphasizes understandability and ease of implementation. PBFT, on the other hand, is designed to tolerate Byzantine faults and ensure consensus even in the presence of malicious nodes.

In summary, the field of Distributed Consensus Algorithms is vital for the development of robust and dependable distributed systems. By addressing challenges related to faults, delays, and uncertainties, these algorithms enable nodes in a distributed environment to collaboratively agree on decisions, thereby ensuring the consistency and reliability of the system as a whole.

# LITERATURE SURVEY

**2.1 OVERVIEW**

In a literary survey, students analyse critically and concisely earlier research and literature related to a particular research problem, and utilize them for their own research purposes. It helps students in understanding the significance of new research and its connections to earlier work.

A literature review is an overview of the previously published works on a topic. The term can refer to a full scholarly paper or a section of a scholarly work such as a book, or an article

**Title :** Based wireless sensor networks for malicious node detection: a survey.

The characteristics of wireless sensor networks and its deployment environment determine that wireless sensor networks is vulnerable to intrusion. In this paper, a malicious nodes detection algorithm based on triangle module fusion operator (MDTMO) is proposed for selective forwarding attack. The algorithm establishes the membership functions according to data packets that received and forwarded by the node, it uses the fusion method of triangle module operator to determine the suspected malicious node. Subsequently, the base station node detects the buffer occupancy and channel occupancy of the suspected malicious node to judge whether the network is in congestion. If the network quality is good, it is considered that the packets loss is due to the selective forwarding attack instead of congestion, so the node is judged to be a malicious node. The experimental results show that compared with the algorithm in [6], MDTMO has improved performance in both the malicious nodes detection rate and false positive rate.

# Title: Cooperative detection for falsification and isolation of malicious nodes through inter-node vote for wireless sensor networks in open environments.

This study presents a novel weighted-trust evaluation based method to detect malicious nodes in wireless sensor networks. Also describes the attacks and security goals in the wireless sensor network, cooperative detection techniques for falsification, isolation of malicious nodes are explained. The aggregation process in WSN is operated in energy efficient way. Data aggregation is performed in every router while forwarding data. It is difficult to identify and isolate the compromised nodes so as to abstain from being deceived by the distorted data infused by the enemy through compromised nodes. In any case, it is trying to secure the flat topology network effectively in light of the poor adaptability and high communication overhead.

# Title: A realtime adaptive trust model based on artificial neural networks for wireless sensor networks.

The present work aims to enhance the routing security in wireless sensor networks (WSNs) against internal attacks, which are mostly launched from compromised nodes. Such nodes would inject malicious behaviour into apparently authentic sensor nodes to disrupt the network’s operation and/or reduce its performance. To this end, an adaptive ambient trust sensor routing (ATSR-ANN) model has been developed by integrating the traditional ATSR model with an artificial neural network (ANN) to provide a real-time capability for avoiding routing through malicious nodes. The problem with the traditional ATSR model is that it uses manually adjusted static parameters to identify malicious nodes. This means that it cannot adapt to changes in the WSN environment, especially the misbehaviour of malicious nodes. Alternatively, the

developed adaptive integrated ATSR-ANN model has a context-aware characteristic thanks to the integrated ANN.

# Title: Privacy preserving secure expansive aggregation with malicious node identification in linear wireless sensor networks

Wireless sensor networks (WSNs), as one of the important technologies in the Internet of +ings (IoT), have been widely deployed to provide practical solutions in various applications, such as environment monitoring, military target sensing, and smart home application. Meanwhile, data privacy leakage in WSNs is becoming the main obstruction, which slows down its further development. For example, in the scenario of a smart home application, videos or pictures collected by wireless IP-cameras could be eavesdropped for illegal profit. As a result, privacy protection on sensitive data is a critical issue that must be addressed in WSNs. In WSNs, the top-k query is one of the critical operations in data aggregation for sensor monitoring process. +e top-k query requests the k lowest or highest data items collected from IoT sensors in WSNs. For example, “collecting the 10 lowest humidity data in forest area A-Z in last 2 hours” is an example of top-k query, which can be performed for fire monitoring. Our aim of this work is to design a secure top-k query approach with privacy-preserving and collusionresisting manners.

**PROBLEM STATEMENT**

Security vulnerabilities in traditional wireless sensor networks (WSNs) pose a threat due to potential malicious node activities. Current detection methods lack robustness. Integrating blockchain with WSNs holds promise, yet a comprehensive survey of existing blockchain-based solutions for malicious node detection is lacking. Addressing this gap is crucial for developing effective, scalable, and secure systems that safeguard WSNs against malicious node. Blockchain integration in wireless sensor networks (WSNs) seeks to bolster security in these distributed systems. WSNs consist of numerous interconnected sensor nodes often vulnerable to attacks due to their decentralized nature. Malicious nodes can jeopardize data integrity and network operations. By incorporating blockchain, a decentralized ledger with immutable records, these networks gain enhanced security and transparency. Blockchain allows nodes to record transactions and interactions, creating an unalterable history of data exchanges. Moreover, smart contracts, programmable protocols within the blockchain, enable the automation of detection mechanisms to identify suspicious activities among nodes. These contracts facilitate the swift isolation and neutralization of potential threats, maintaining the network's integrity. This integration not only fortifies security but also provides continuous surveillance and threat mitigation within the WSN. It ensures the reliability of data transmission, safeguarding sensitive information. However, challenges persist, including the computational overhead in implementing blockchain in resource-constrained sensor nodes and the scalability of such a system. Despite these hurdles, the integration of blockchain in WSNs holds promise in addressing security concerns and enabling robust, trustworthy communication among sensor nodes

**METHODOLOGY**

# EXISTING METHODOLOGY

Wireless Sensor Networks (WSNs) are broadly applied for various applications in tracking and surveillance due to their ease of use and other distinctive characteristics compelled by real-time cooperation among the sensor nodes. In WSNs, security is becoming a critical issue, as the techniques for malicious node detection adopt a one-time, centralized decision-making approach. With this paradigm, errors are difficult to avoid, and reproducibility and traceability are challenging. Hence, malicious node discovery technologies in conventional WSNs cannot assure traceability and fairness of the detection method. Herein, this paper discusses an in-depth survey of a block chain-based approach for malicious node detection, an exhaustive examination of the integration of block chain techniques with WSNs (BWSN), and insights into this novel concept. This survey discusses the architecture, sector-wise applications, and uses of BWSN. Moreover, this survey describes malicious node detection based on BWSN in two parts: 1) the BWSN architecture for detecting the malicious nodes and 2) the smart contract aspects in malicious node detection. Next, this survey explains the contributions of block chain for WSN data management, which involves online information aggregation and may include auditing, event logs, and storage for information analysis and offline query processing. This survey first presents the conventional WSN solutions then the block chain-based WSN solutions for data management. Additionally, this survey discusses the contributions of block chain for WSN security management. It first examines the centralized WSN models for security problems, followed by a discussion of the block chain-based WSN solutions for security management, such as offering access control, preserving information integrity, guaranteeing privacy, and ensuring WSNs’ node longevity.

# PROPOSED METHODOLOGY

Underlined the importance of formulating/defining features and converting the received signals into those features. Characterized the impact of selecting kernel functions and k-SVM parameters. Quantified the impact of both the training and testing dataset length on the accuracy. We have shown that, the accuracy of the linear-kernel based, RBF-kernel-based, and polynomial- kernel-based k-SVM classifiers can be improved. The impact of selecting γ in relation to the accuracy has been presented, especially in the case of k-SVM. Identification of the attacks is done by using the svm.

# ADVANTAGES

▶ k-SVM is a powerful machine learning algorithm that can provide high accuracy in detecting malicious nodes in WSNs.

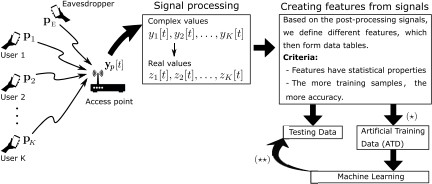
▶ k-SVM is a robust algorithm that can handle non-linear and high- dimensional data, making it suitable for WSN applications.

▶ k-SVM can generalize well to unseen data, making it suitable for real- world applications where new types of attacks may appear

▶ k-SVM can scale well to large datasets, making it suitable for large WSNs with many nodes.

▶ The results of k-SVM can be easily interpreted, allowing for easy analysis and understanding of the malicious node detection process.

# BLOCK DIAGRAM



**DESCRIPTION OF MODULES**

**CLUSTER HEAD**

In cluster head there are three functions which is at cluster head ID, position and energy, in cluster member for models which is a cluster member id position energy under cluster id which the cluster member. In the message module there are certain categories which are cluster ID cluster head position cluster head energy cluster member id on the cluster member position and the message and the trust evaluation. There are certain functionalities which is the base station level of cyber-attack detection. The trust evaluation is done in the base station where the detection function is also maintained.

# SENSOR NODE MODULE

The second module consists of a sensor node where the number of cluster heads and the number of cluster members can be created in cluster member. The member information will be displayed where the member is identified in which cluster at the number of member id the position of member id and the energy all of them can be connected with the base station.

# MESSAGING MODULE

When the cluster member sends a message, the message can be viewed in the cluster head form. As many number of cluster member can send a message all of them will be observed in the cluster head SN organize a group of clusters where the base station node and the CH are trustworthy and not compromised by any attack.

# SOFTWARE REQUIREMENT SPECIFICATION

## HARDWARE REQUIREMENTS

* + - System : Pentium Core 2 Duo
    - Hard Disk : 500 GB.
    - Monitor : 15 VGA Colour.
    - Mouse : Logitech.
    - Ram : 2 GB DDR2 RAM.
    - Keyboard : LG 104 Keys Keyboard.

## SOFTWARE REQUIREMENTS

* + - Operating System : Windows 7 or Windows 11.
    - Coding Language : Java, XML,OS windows 11.
    - IDE : VSCode or netBeans.

# SYSTEM DESIGN

**ARCHITECTURE DIAGRAM**

CLUSTER HEAD

Base station

Message sharing

Sybil attack detection



Result

DV-HOP WITH K-SVM

# FRONT END: JAVA

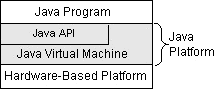
The software requirement specification is created at the end of the analysis task. The function and performance allocated to software as part of system engineering are developed by establishing a complete information report as functional representation, a representation of system behaviour, an indication of performance requirements and design constraints, appropriate validation criteria.

# FEATURES OF JAVA

Java platform has two components:

* + The *Java Virtual Machine* (Java VM)
  + The *Java Application Programming Interface* (Java API) The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries (*packages*) of related components.

The following figure depicts a Java program, such as an application or applet, that's running on the Java platform. As the figure shows, the Java API and Virtual Machine insulates the Java program from hardware dependencies.



As a platform-independent environment, Java can be a bit slower than native code. However, smart compilers, well-tuned interpreters, and just-in-time byte code compilers can bring Java's performance close to that of native code without threatening portability.

# SOCKET OVERVIEW:

A network socket is a lot like an electrical socket. Various plugs around the network have a standard way of delivering their payload. Anything that understands the standard protocol can “plug in” to the socket and communicate.

Internet protocol (IP) is a low-level routing protocol that breaks data into small packets and sends them to an address across a network, which does not guarantee to deliver said packets to the destination.

Transmission Control Protocol (TCP) is a higher-level protocol that manages to reliably transmit data. A third protocol, User DatagramProtocol (UDP), sits next to TCP and can be used directly to support fast, connectionless, unreliable transport of packets.

# CLIENT/SERVER:

A server is anything that has some resource that can be shared. There are compute servers, which provide computing power; print servers, which manage a collection of printers; disk servers, which provide networked disk space; and

web servers, which store web pages. A client is simply any other entity that wants to gain access to a particular server.

A server process is said to “listen” to a port until a client connects to it. A server is allowed to accept multiple clients connected to the same port number, although each session is unique. To manage multiple client connections, a server process must be multithreaded or have some other means of multiplexing the simultaneous I/O.

# RESERVED SOCKETS:

Once connected, a higher-level protocol ensues, which is dependent on which port user are using. TCP/IP reserves the lower, 1,024 ports for specific protocols. Port number 21 is for FTP, 23 is for Telnet, 25 is for e-mail, 79 is for finger, 80 is for HTTP, 119 is for Netnews-and the list goes on. It is up to each protocol to determine how a client should interact with the port.

# JAVA AND THE NET:

Java supports TCP/IP both by extending the already established stream I/O interface. Java supports both the TCP and UDP protocol families. TCP is used for reliable stream-based I/O across the network. UDP supports a simpler, hence faster, point-to-point datagram-oriented model.

# INETADDRESS:

The InetAddress class is used to encapsulate both the numerical IP address and the domain name for that address. User interacts with this class by using the name of an IP host, which is more convenient and understandable than its IP address. The InetAddress class hides the number inside. As of Java 2, version 1.4, InetAddress can handle both IPv4 and IPv6 addresses.

# FACTORY METHODS:

The InetAddress class has no visible constructors. To create an InetAddress object, user use one of the available factory methods. Factory

methods are merely a convention whereby static methods in a class return an instance of that class. This is done in lieu of overloading a constructor with various parameter lists when having unique method names makes the results much clearer.

Three commonly used InetAddress factory methods are:

1. Static InetAddressgetLocalHost ( ) throws UnknownHostException
2. Static InetAddressgetByName (String hostName) throwsUnknowsHostException
3. Static InetAddress [ ] getAllByName (String hostName) throwsUnknownHostException

# INSTANCE METHODS:

The InetAddress class also has several other methods, which can be used on the objects returned by the methods just discussed. Here are some of the most commonly used.

Boolean equals (Object other)- Returns true if this object has the same Internet address as other.

1. byte [ ] get Address ( )- Returns a byte array that represents the object’s Internet address in network byte order.
2. String getHostAddress ( ) - Returns a string that represents the host address associated with the InetAddress object.
3. String get Hostname ( ) - Returns a string that represents the host name associated with the InetAddress object.
4. booleanisMulticastAddress ( )- Returns true if this Internet address is a multicast address. Otherwise, it returns false.
5. String toString ( ) - Returns a string that lists the host name and the IP address for convenience.

# TCP/IP CLIENT SOCKETS:

TCP/IP sockets are used to implement reliable, bidirectional, persistent, point-to-point and stream-based connections between hosts on the Internet. A socket can be used to connect Java’s I/O system to other programs that may reside either on the local machine or on any other machine on the Internet.

There are two kinds of TCP sockets in Java. One is for servers, and the other is for clients. The Server Socket class is designed to be a “listener,” which waits for clients to connect before doing anything. The Socket class is designed to connect to server sockets and initiate protocol exchanges.

The creation of a Socket object implicitly establishes a connection between the client and server. There are no methods or constructors that explicitly expose the details of establishing that connection. Here are two constructors used to create client sockets

Socket (String hostName, intport) - Creates a socket connecting the local host to the named host and port; can throw an UnknownHostException or anIOException. Socket (InetAddressipAddress, intport) - Creates a socket using a preexistingInetAddressobject and a port; can throw an IOException.

A socket can be examined at any time for the address and port information associated with it, by use of the following methods:

* InetAddressgetInetAddress ( ) - Returns the InetAddress associated with the Socket object.
* IntgetPort ( ) - Returns the remote port to which this Socket object is connected.
* IntgetLocalPort ( ) - Returns the local port to which this Socket object is connected.

Once the Socket object has been created, it can also be examined to gain access to the input and output streams associated with it. Each of these methods can throw an IO Exception if the sockets have been invalidated by a loss of connection on the Net.

Input Streamget Input Stream ( ) - Returns the InputStream associated with the invoking socket.

Output Streamget Output Stream ( ) - Returns the OutputStream associated with the invoking socket.

# TCP/IP SERVER SOCKETS:

Java has a different socket class that must be used for creating server applications. The ServerSocket class is used to create servers that listen for either local or remote client programs to connect to them on published ports. ServerSockets are quite different form normal Sockets.When the user create a ServerSocket, it will register itself with the system as having an interest in client connections.

* ServerSocket(int port) - Creates server socket on the specified port with a queue length of 50.
* Serversocket(int port, int maxQueue) - Creates a server socket on the specified portwith a maximum queue length of maxQueue.
* ServerSocket(int port, int maxQueue, InetAddress localAddress)-Creates a server socket on the specified port with a maximum queue length of maxQueue. On a multihomed host, localAddress specifies the IP address to which this socket binds.
* ServerSocket has a method called accept( ) - which is a blocking call that will wait for a client to initiate communications, and then return with a normal Socket that is then used for communication with the client.

# URL:

The Web is a loose collection of higher-level protocols and file formats, all unified in a web browser. One of the most important aspects of the Web is that Tim Berners-Lee devised a saleable way to locate all of the resources of the Net. The Uniform Resource Locator (URL) is used to name anything and everything reliably.

The URL provides a reasonably intelligible form to uniquely identify or address information on the Internet. URLs are ubiquitous; every browser uses them to identify information on the Web.

# CODING

# NetworkController

# Main.java

# /\*

# \* To change this license header, choose License Headers in Project Properties.

# \* To change this template file, choose Tools | Templates

# \* and open the template in the editor.

# \*/

# package networkcontroller;

# import de.javasoft.plaf.synthetica.SyntheticaBlueLightLookAndFeel;

# import javax.swing.UIManager;

# /\*\*

# \*

# \* @author Elcot

# \*/

# public class Main {

# public static void main(String[] args)

# {

# try

# {

# UIManager.setLookAndFeel(new SyntheticaBlueLightLookAndFeel());

# 

# NCFrame sf=new NCFrame();

# sf.setTitle("Network Controller");

# sf.setVisible(true);

# sf.setResizable(false);

# 

# NCReceiver sr=new NCReceiver(sf);

# sr.start();

# }

# catch (Exception ex)

# {

# //System.out.println(ex);

# }

# }

# }

# VANETNodes

# Main.java

# /\*

# \* To change this license header, choose License Headers in Project Properties.

# \* To change this template file, choose Tools | Templates

# \* and open the template in the editor.

# \*/

# package vanetnodes;

# import de.javasoft.plaf.synthetica.SyntheticaBlueLightLookAndFeel;

# import javax.swing.JFrame;

# import javax.swing.JOptionPane;

# import javax.swing.UIManager;

# /\*\*

# \*

# \* @author Elcot

# \*/

# public class Main

# {

# public static void main(String[] args)

# {

# try

# {

# UIManager.setLookAndFeel(new SyntheticaBlueLightLookAndFeel());

# 

# int nodeid=Integer.parseInt((JOptionPane.showInputDialog(new JFrame(),"Enter the Vehicle Id:")).trim());

# 

# NodeFrame nf=new NodeFrame(nodeid);

# nf.setTitle("Vehicle - "+nodeid);

# nf.jLabel1.setText("Vehicle - "+nodeid);

# nf.setVisible(true);

# nf.setResizable(false);

# 

# NodeReceiver nr=new NodeReceiver(nf,nodeid);

# nr.start();

# }

# catch (Exception ex)

# {

# //System.out.println(ex);

# }

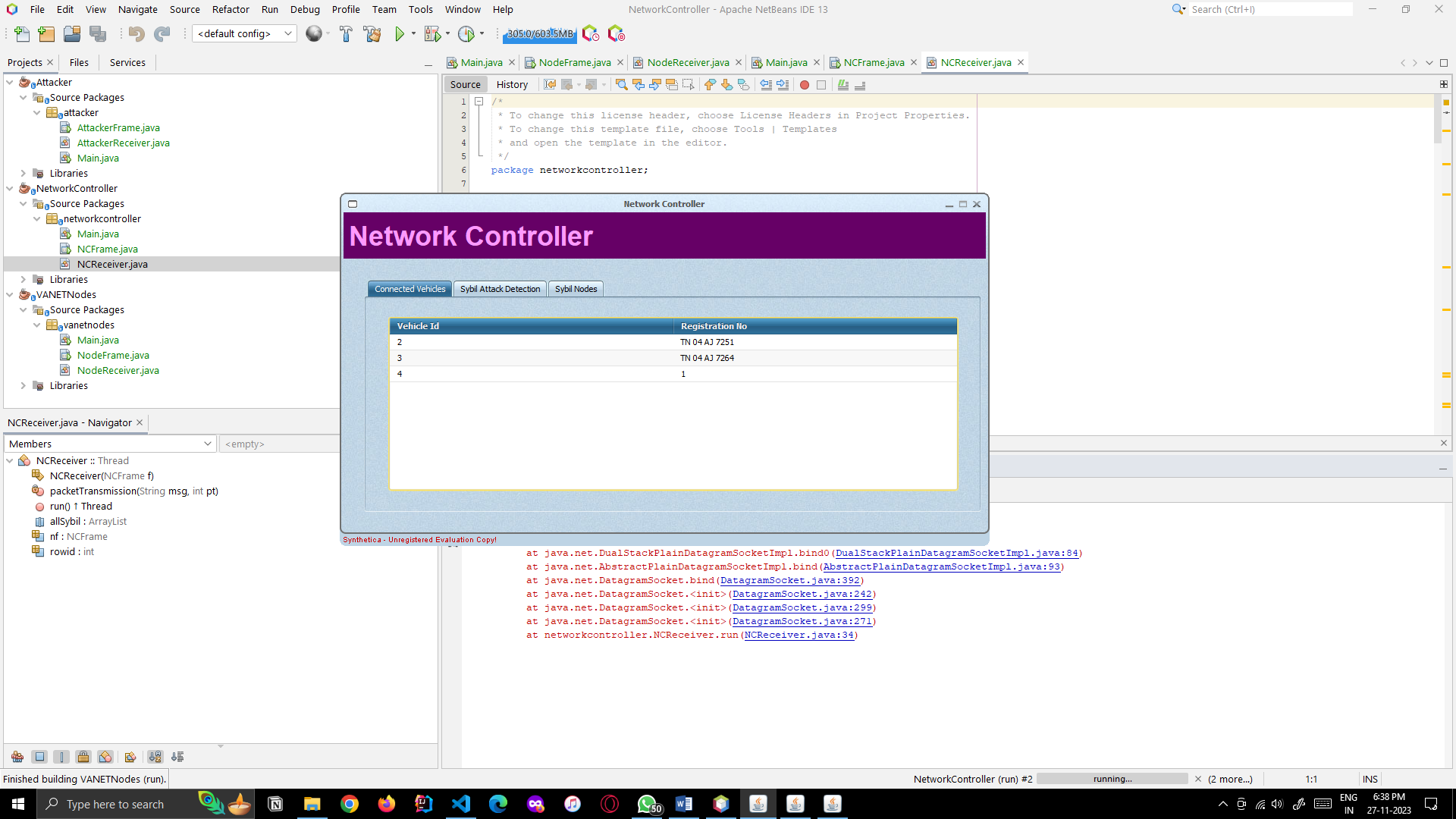
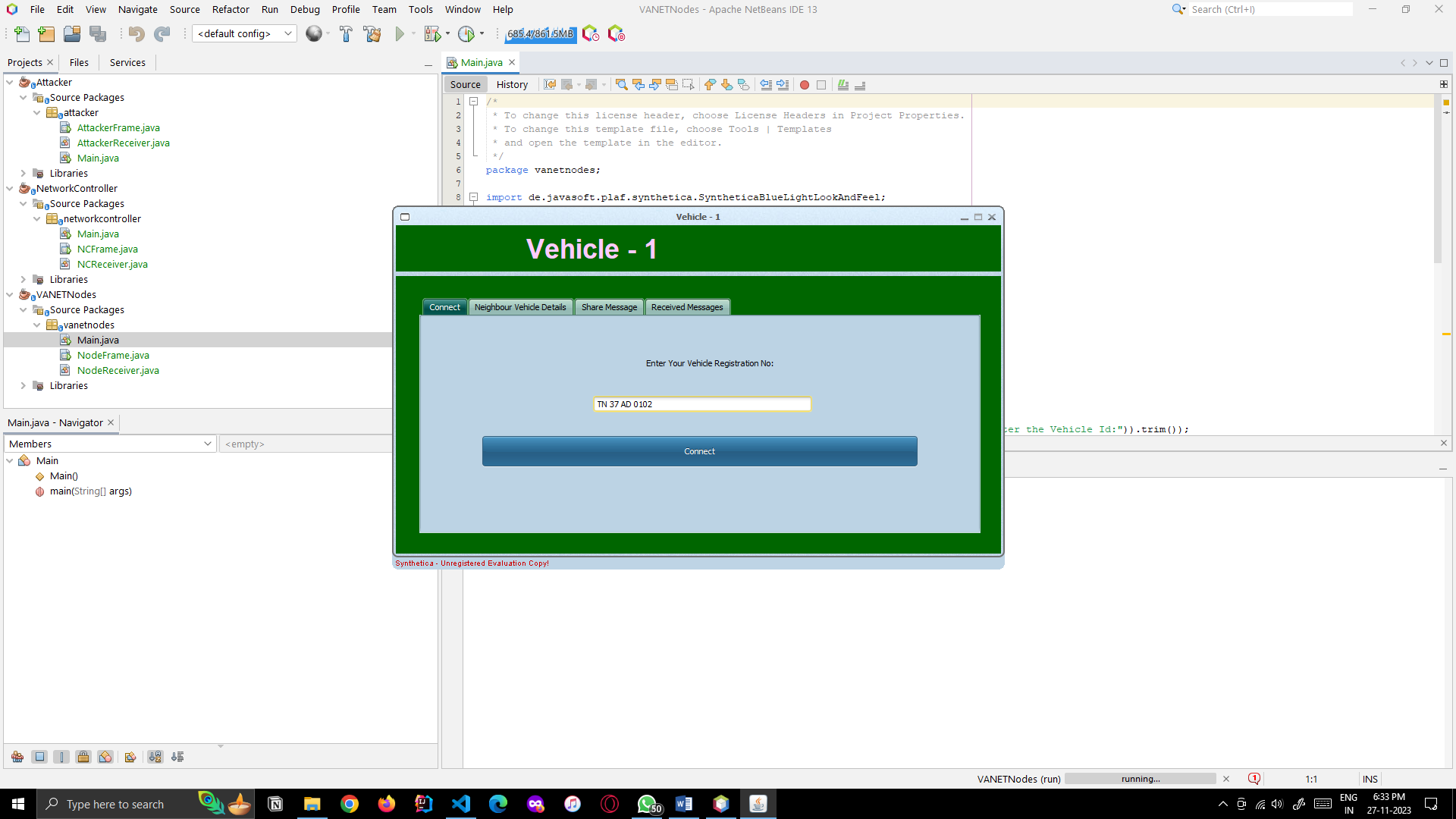
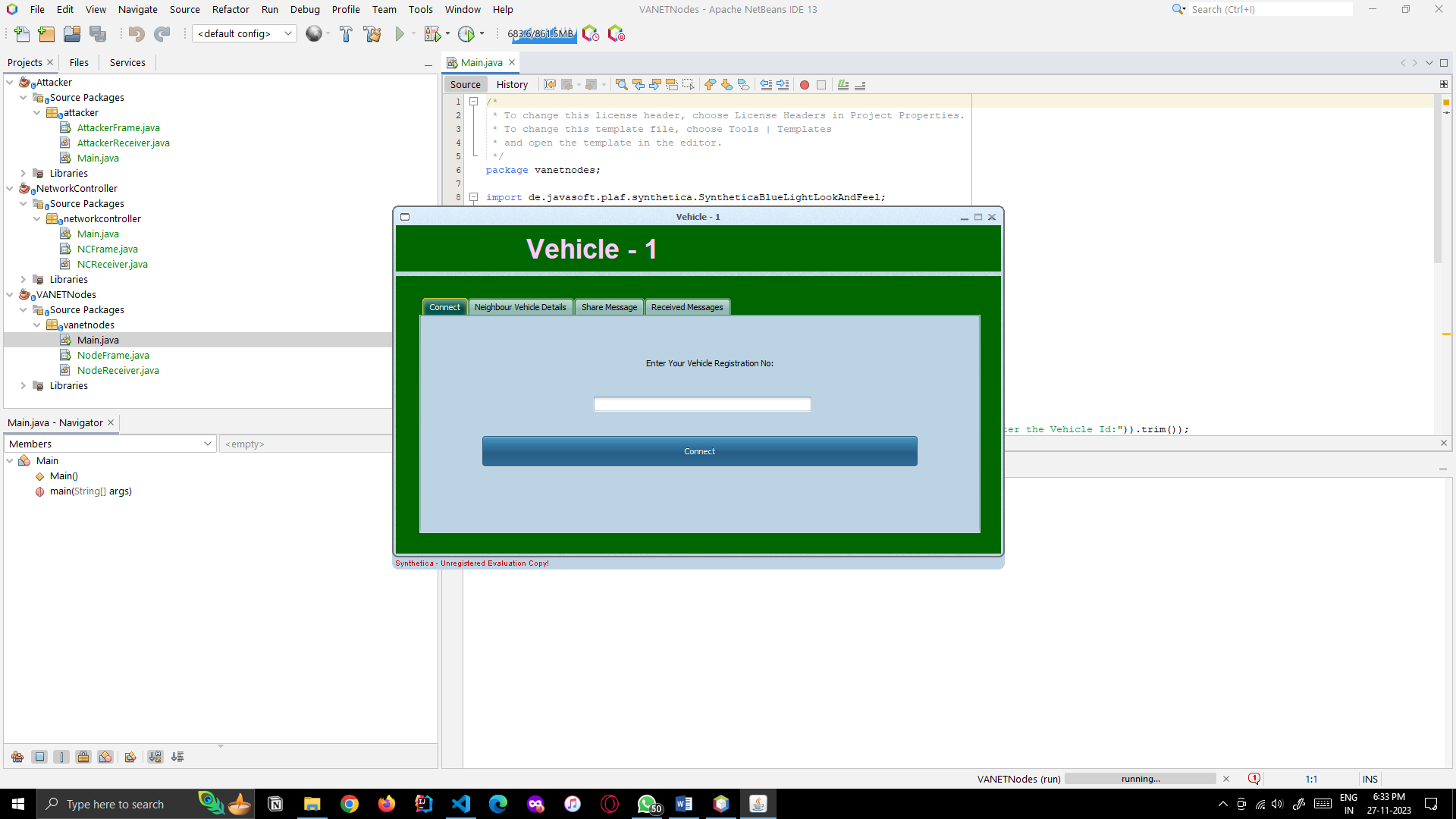
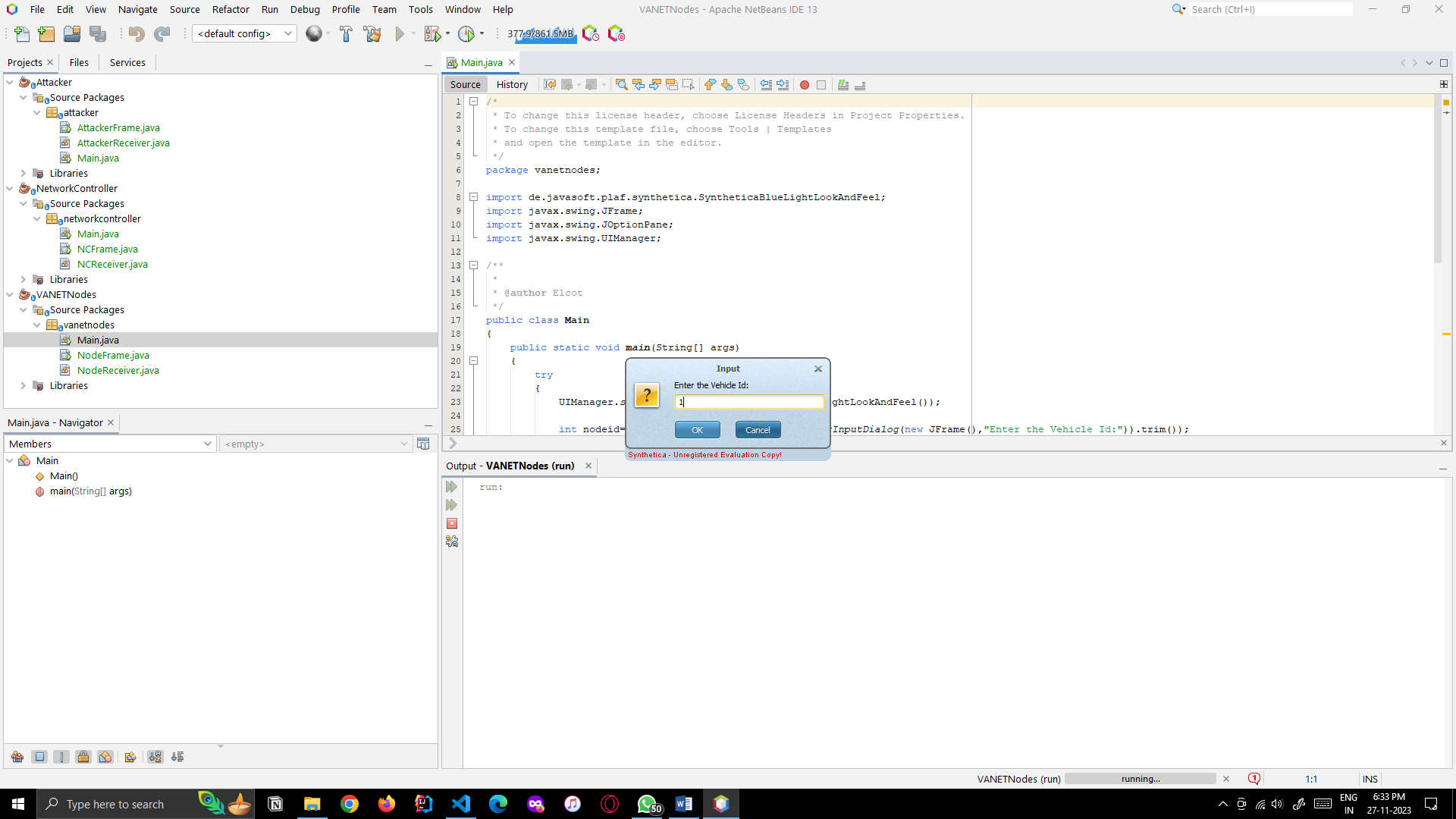
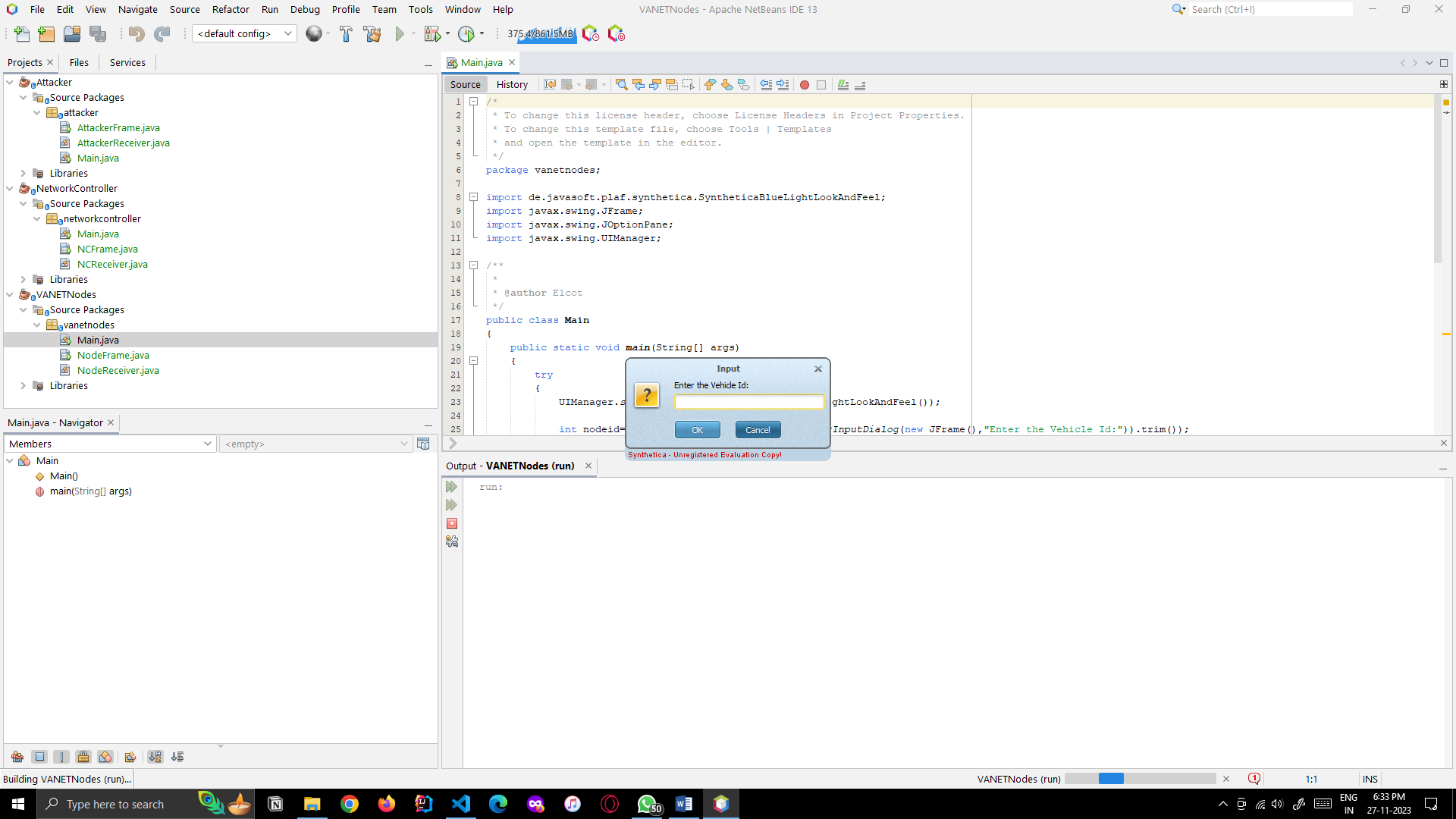
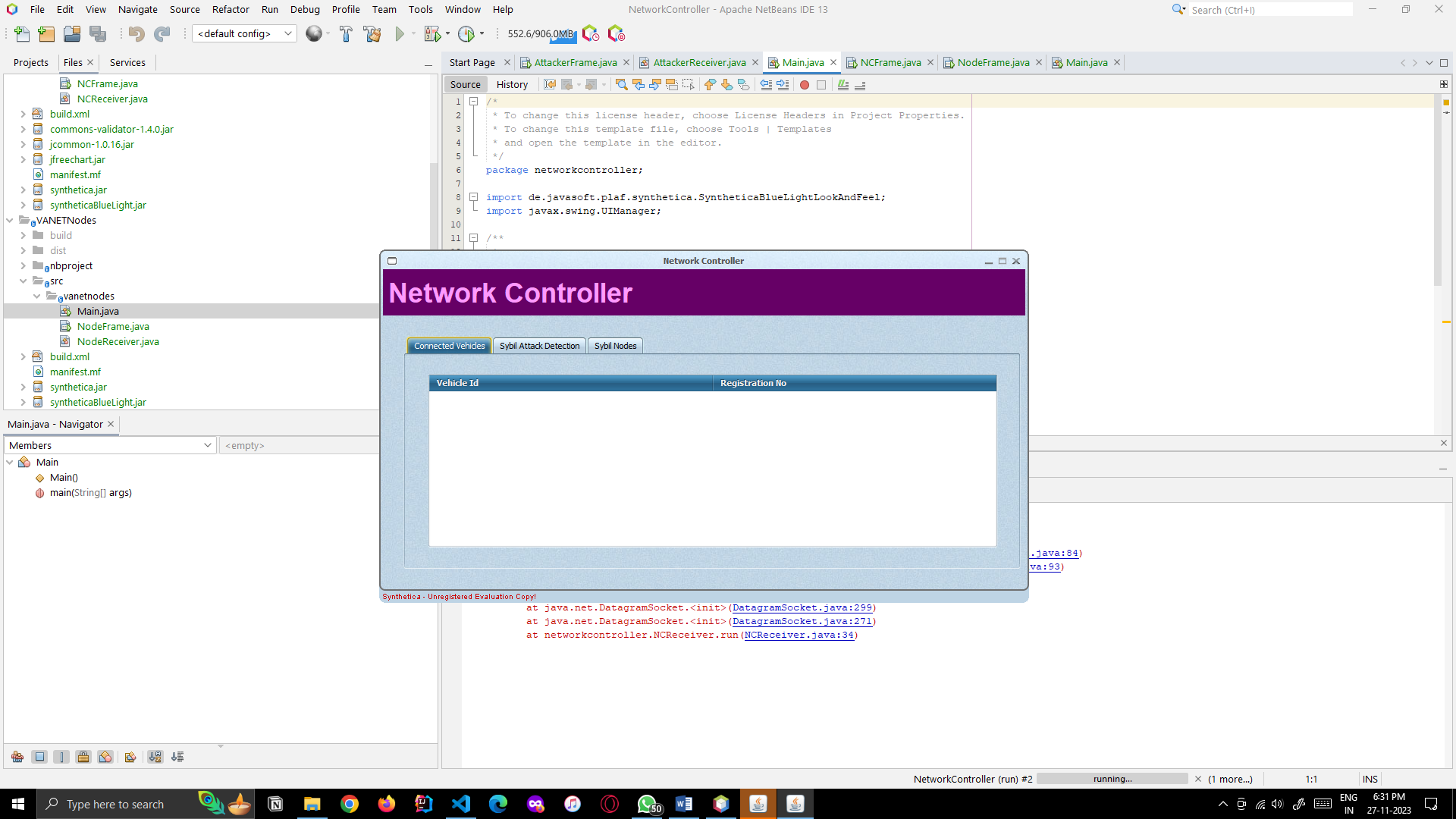
# }

# }

**REFERENCES**

|  |  |
| --- | --- |
| [1] | L. Min and G. Ranxin, “Malicious nodes detection algorithm based on triangle module fusion operator in wireless sensor networks,” in 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Dec. 2019, vol. 1, pp. 118–  121, DOI: 10.1109/IAEAC47372.2019.8997710. |
| [2] | Y. Kimura, E. Nii, and Y. Takizawa, “Cooperative Detection for Falsification and Isolation of Malicious Nodes through Inter-node Vote for Wireless Sensor Networks in Open Environments,” in 2019 Global Information Infrastructure and Networking Symposium (GIIS), Dec.  2019, pp. 1–3, DOI: 10.1109/GIIS48668.2019.9044952. |
| [3] | B. Jaint, V. Singh, L. K. Tanwar, S. Indu, and N. Pandey, “A Realtime Adaptive Trust Model Based on Artificial Neural Networks for Wireless Sensor Networks,” in 2018 2nd IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES), Oct.  2018, pp. 1183–1187, DOI: 10.1109/ICPEICES.2018.8897307. |
| [4] | Shah K, Jinwala D. EPCT: An Efficient Privacy-Preserving and Collusion-Resisting Top-k Query Processing in WSNs Privacy- preserving secure expansive aggregation with malicious node identification in linear wireless sensor networks. Frontiers of Computer  Science. 2021 Dec;15(6):1-9. |
| [5] | A. J. Manuel, G. G. Deverajan, R. Patan, and A. H. Gandomi, “Optimization of Routing-Based Clustering Approaches in Wireless Sensor Network: Review and Open Research Issues,” Electronics, vol. 9,  no. 10, Art. no. 10, Oct. 2020, DOI: 10.3390/electronics9101630. |

**OUTPUT:**

****