IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

DETECTION OF FALSIFICATION ATTACKS IN VANET USING MACHINE LEARNING

Ms. PREETHA V[1], RUDRAPRIYA M[2], SHARLI IRIS JOHN BASKAR[3], MOUNICA M[4]

[1]*Assistant Professor, Department of CSE, Sri Ramakrishna Institute of Technology, Coimbatore ,Tamil Nadu, India*

[2,3,4]*Student, Department of CSE, Sri Ramakrishna Institute of Technology, Coimbatore ,Tamil Nadu, India*

Abstract: The **Vehicle Data Prediction** is an innovative web-based application designed to classify vehicle movement data as either **legitimate or falsified** using **Machine Learning**. With the increasing need for reliable traffic monitoring, logistics optimization, and fraud detection, this system provides a user-friendly interface where users can input key vehicle parameters such as **position, speed, and acceleration**. The entered data is then processed through two powerful machine learning models **Logistic Regression** and **Random Forest Classifier** which yields accuracy of 77.63% and 89.08% which analyze and predict the authenticity of the data.

Key words: Machine Learning Algorithms; Random Forest; Logistic Regression; Web Application

I. INTRODUCTION

In today's fast-paced world, vehicle tracking and data analysis play a crucial role in ensuring security, efficiency, and accuracy in various industries, from transportation and logistics to fraud detection. This project introduces a Vehicle Data Prediction System, a web-based application that uses Machine Learning models to analyze vehicle movement data and predict whether the data is legitimate or falsified. By leveraging Logistic Regression and Random Forest Classifier, the system takes key parameters such as position, speed, and acceleration to assess vehicle behavior. A user-friendly web interface allows users to enter the vehicle data, submit it for prediction, and receive instant results. This research offers an interactive and effective way to identify irregularities in vehicle movement by showcasing how machine learning can be smoothly incorporated into a web-based prediction tool which ensures security.

II. METHODOLOGY

The Vehicle Data Prediction System uses a structured process that blends machine learning and web development to create an accurate and user-friendly prediction tool. In order to guarantee consistency in predictions, the process starts with data preprocessing, where vehicle movement parameters like position, speed, and acceleration are standardized using a scalar model. In preprocessing entails the necessary data preparation procedures prior to model training. It involves handling missing data, eliminating outliers, choosing pertinent features, scaling features to a consistent range, converting categorical data into numerical representations, dividing data for testing and training, correcting class imbalance, and lowering dimensionality. Preprocessing data effectively guarantees that it is clean, appropriate for modeling, and optimizes model performance. Two Machine Learning models Logistic Regression and Random Forest Classifier are used to classify whether the given vehicle data is Legitimate or Falsified. These models are trained on historical data, enabling them to detect anomalies and patterns effectively. The prediction process involves taking user input, transforming it into a format suitable for model evaluation, and generating results instantly. The Flask framework is used to develop the backend API, which handles the prediction requests. The front end is built using HTML, CSS, and JavaScript, featuring an interactive and visually appealing user interface with animations and glowing effects.

When users enter the vehicle details and submit them, the system processes the data, makes predictions, and displays the results dynamically on a new page with animated text effects. To enhance user experience, a reset button allows users to clear input fields and enter new values quickly. The entire methodology ensures seamless interaction between the Machine Learning models and the Web Interface, providing real-time and accurate vehicle data predictions.

III. PROPOSED SYSTEM

The methodology has three major stages are Dataset Extraction, Data Preparation and Classification. There are 225 simulations with various traffic circumstances in the VeReMi dataset. There are two kinds of files in every simulation: log files and truth file. In a simulation, there is only one ground file that contains the real behavior of a vehicle in the network. To distinguish a bad car from a bad one, the attacker mode is also included in the ground truth data. However, in the simulation, the number of data logs is equal to the traffic in the network. Each vehicle generates a log containing all BSMs received from other vehicles. The log file contains false information because, similar to the anti-fraud site, the attacker sent false messages from the BSM. The creation of a labeled dataset requires the joining of truth and log files. The truth file is assigned to each simulation's log file during the data extraction phase. Since there are as many log files for a single simulation as there are receivers, combining these disparate log files into a single file is the initial stage. A distinct ID called messageID is present in both log and ground truth files. The data in the combined log file must be mapped to the attacker type in the ground truth file in order to produce a labeled dataset. This procedure is carried out five times because the VeReMi dataset contains five distinct seeds of the same situation, which are used to introduce randomization into the network. In order to get a merged dataset for a single scenario, all five repetitions were eventually integrated.

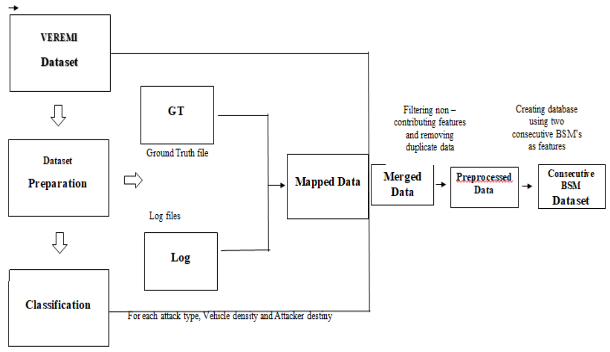

Fig.1.Architecture of Proposed System

IV. EXISTING SYSTEM

Conventional falsification detection systems use preset thresholds and rules to identify data irregularities. They might, for example, indicate any communication that has data (such position or speed) that is outside of normal ranges or that differs noticeably from prior messages that have been received. Due to their relative simplicity, these systems may not be able to handle complex falsification techniques or changing assault patterns. Due to their inability to learn from or adjust to previous events, they may also have significant false-positive rate.

Using Machine Learning (ML) algorithms to identify anomalies in VANETs has become popular. These systems look for odd or suspicious behavior that might point to falsification by analyzing data patterns. Using the VeReMi dataset, numerous researchers have developed a misbehavior detection framework to identify position falsification assaults. In order to train the model, some of the present effort entails adding calculating features such a change in Position and Speed.

This suggested approach eliminates the network's vehicle-to-vehicle dependence because RSU provides a more comprehensive understanding of any network misconduct. For the attacker's vehicle to be detected, it must be within sight of other legal vehicles. The scheme identifies attacks faster than the current method of linking a transmitter and receiver by pairing a vehicle with a Roadside Unit (RSU). Using the vehicle-RSU combination, the system can detect the attacking vehicle before it has a chance to impersonate a legitimate vehicle. Current methods rely on the network's genuine vehicles to identify the assault. Transmitter-receiver matching methods often fail to detect cyber attackers until the legitimate tool reaches the attacker's tool. In comparison, the tool-RSU combination can provide more protection by identifying the attack regardless of the number of attackers in the network.

V. SYSTEM DESIGN

The process begins with Vehicle Data Input, where key parameters such as position (posx, posy), speed (spdx, spdy), and other movement metrics are collected. This raw data undergoes Data Preprocessing, which includes essential steps like Cleaning, Scaling, and Encoding to ensure consistency and improve the accuracy of predictions.

Once preprocessed, the data is passed to the ML Model Classification stage, where machine learning algorithms such as Random Forest and Logistic Regression analyze the input features and predict whether the data is authentic or fraudulent. These trained models are then integrated into a Real-Time API, developed using Flask or FlaskAPI, allowing seamless interaction between the frontend user interface and backend processing. Finally, the prediction results are generated and displayed to the user, indicating whether the vehicle data is Legitimate (0) or Malicious (1). The output is presented in a visually appealing and interactive UI, incorporating animations and a reset option for users to enter new data for further predictions.

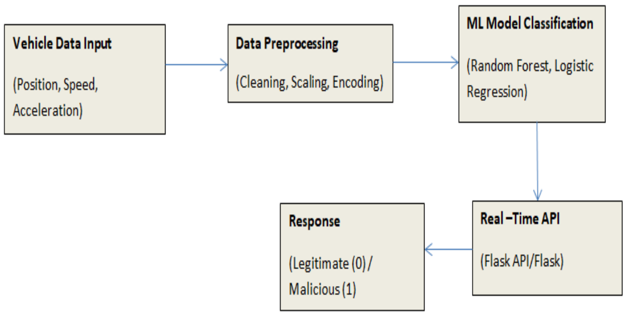

Fig.2.System Architecture

VI. TECHNIQUES USED

A. Data Preprocessing:

In machine learning, preprocessing entails the necessary data preparation procedures prior to model training. It involves handling missing data, eliminating outliers, choosing pertinent features, scaling features to a consistent range, converting categorical data into numerical representations, dividing data for testing and training, correcting class imbalance, and lowering dimensionality. Preprocessing data effectively guarantees that it is clean, appropriate for modeling, and optimizes model performance.

B. Random Forest Algorithm:

One popular machine learning approach is Random Forest approach. It is an algorithm for Supervised Machine learning. It can be used to solve machine learning challenges that combine classification and regression. The random forest algorithm is based on the concept of ensemble learning, involves in combining multiple classifiers to improve the model's performance and solve an issue that is challenging. The Random Forest Algorithm makes use of a number of decision tree-based classifiers. Each decision tree is first trained on an individual basis. Based on these trees' predictions, the random forest then predicts the average of these results.

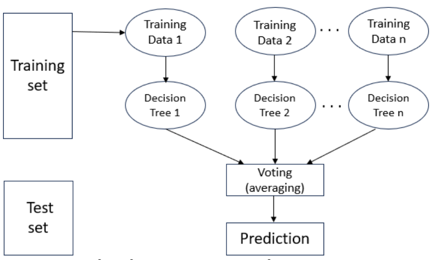

Fig.3.Working of Random Forest

C. Logistic Regression:

Logistic Regression is a method used to estimate the probability of an event occurring, usually when the outcome is one of two possible choices (e.g. yes/no or true/false). It works by finding a relationship between input data (age, income, etc.) and the probability of occurrence. Instead of giving a clear answer like 0 or 1, the model gives a probability between 0 and 1 that indicates how likely the outcome is. For example, it will predict the probability as 0.75, meaning there is a 75% chance of the event occurring. In this way, it uses an 'S' shaped curve (called the logistic function) instead of a straight line, which helps the model predict two possible outcomes (0 or 1) as information. The predicted values form an 'S' shaped curve, which always falls between the value 0 and 1. This curve is called the logistic function or the Sigmoid function. The model uses a threshold value to decide whether the outcome should be 0 or 1. For instance, if the value is below the threshold, the result will likely be 0; if it's above the threshold, the result will likely be 1. This threshold helps classify the data into one of two categories.

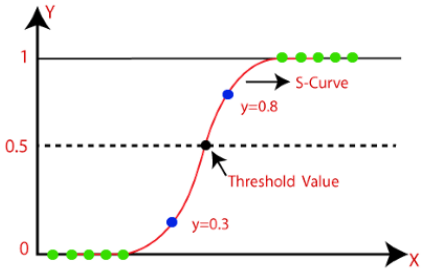

Fig.4.Working of Logistic Regression

D. REST API:

REST API (Representational State Transfer API) acts as a communication bridge between the frontend user interface and the backend machine learning models. It is a way for different systems to communicate over the internet using standard HTTP methods like GET, POST, PUT, and DELETE. It follows REST principles, making it lightweight, scalable, and platform-independent. It enables seamless data exchange, allowing users to send vehicle movement data and receive predictions on whether the data is legitimate or falsified**.** Flask is a lightweight Python web framework used to build web applications and APIs. It is simple, flexible, and widely used for response is then displayed dynamically with an engaging visual effect. Overall, this frontend design ensures a user-friendly, intuitive, and aesthetically pleasing experience, making it easy for users to

creating REST APIs in machine learning and data science projects.

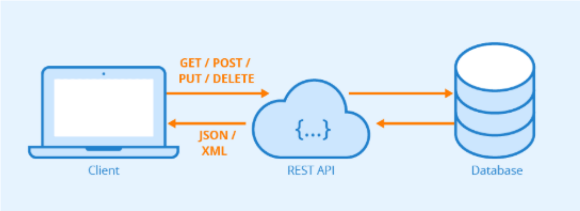

Fig.5.Working of REST API

E. Frontend:

The frontend of this project is designed to provide an interactive and visually appealing user experience using HTML, CSS, and JavaScript. The user interface allows individuals to input vehicle-related data such as position, speed, and acceleration, which is then processed by the backend for prediction. The form is structured with responsive input fields and a submit button that sends data to the Flask backend through a POST request. A reset button allows users to clear the fields and enter new values seamlessly. This frontend effectively communicates with the Flask backend, where the machine learning model processes the data and returns predictions. The

interact with the application and receive real-time vehicle data predictions effortlessly.

VII. APPLICATIONS

- Vehicle Fraud Detection: Identifies whether vehicle data is falsified or legitimate, helping in fraud prevention.
- Autonomous Vehicle Validation**:** Ensures the accuracy of sensor data in self-driving cars to improve navigation and safety.
- Smart City Development**:** Integrates with IoT-based traffic management systems for efficient urban planning and road safety.
- Fleet Management**:** Helps logistics and transportation companies monitor vehicle data for optimized fleet performance.
- Traffic Monitoring: Assists in analyzing real-time vehicle movement and detecting anomalies in traffic data.
- Law Enforcement & Security**:** Aids police and security agencies in identifying suspicious vehicle behavior and preventing illegal activities.

VIII. RESULT ANALYSIS

The performance of this project is evaluated based on the accuracy and efficiency of predictions made by the machine learning models. The system processes vehicle data, including position, speed, and acceleration, and determines whether the data is legitimate or falsified. The Random Forest and Logistic Regression models are used for classification, and their accuracy is assessed using standard metrics like precision, recall, and F1-score. The implementation of a real-time API using Flask ensures quick and scalable data processing, enabling seamless interaction with the system. Additionally, the frontend is designed to be user-friendly, with an interactive and visually appealing interface that enhances usability. The system also incorporates error-handling mechanisms to manage missing or incorrect inputs, making it more robust.

Algorithms used	**Accuracy**
Logistic Regression	77.63%
Random Forest	89.08%

TABLE.1. ACCURACY OBTAINED FOR VeReMi DATASET

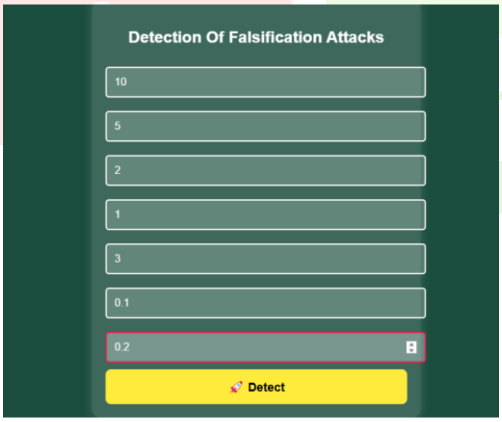

Fig.6.User Input

Fig.6. Prediction Result

IX. CONCLUSION

This project effectively illustrates how to categorize vehicle data as authentic or fraudulent by integrating machine learning models with a real-time REST API. The system guarantees accurate and dependable predictions by utilizing Random Forest and Logistic Regression models in conjunction with efficient data preprocessing methods. The backend and frontend may interact seamlessly when using Flask for API development, which makes it practical and effective for real-world applications. Furthermore, the user-friendly design improves usability by giving users a seamless experience. The project may find use in real-time traffic monitoring, intelligent transportation systems, and fraud detection. All things considered, this solution demonstrates the effectiveness and scalability of machine learning in automation and decision-making for vehicle data validation.

REFERENCES

[1] Bala Manoj Kumar P, Srinivasa Perumal R , Nadesh R K, Arivuselvan K ,"Type Diabetes mellitus prediction using Deep Neural Networks Science and Computational Intelligence 2022, ID: 216 (2023) 21–30.

[2] Muhammad Exell Febriana, Fransiskus Xaverius Ferdinana, Gustian Paul Sendania, Kristien Margi Suryanigrum, Rezki Yunanda," Diabetes prediction using supervised machine learning",7th International Conference on Computer classifier",International Journal of Cognitive Computing in Engineering 1 (2020) 55–61.

[3] MD. Shafiqul Islam, Marwa K. Garaqe (Member IEEE), Samir Brahim Belhaouari (Senior Member,IEEE), and Muhammad A. Abdul-Ghani "Advanced Techniques for Predicting the Future Progression of Type 2 Diabetes",IEEEACCESS.2020.3005540,VOLUME 8, 2020

[4] Varun Sapra, Luxmi Sapra, Akashdeep Bhardwaj, Salil Bharany, Akash Saxena, Faten Khalid Karim, Sara Ghorashi, Ali Wagdy Mohamed," Integrated approach using

deep neural network and CBR for detecting severity of coronary artery disease ", doi:10.1016/j.aej.2023.01.029 (2023) 1110-0168.

[5] Yue Huang, YingBo Ren, Hai Yang, YiJie Ding, Yan Liu, YunChun Yang, AnQiong Mao, Tan Yang, YingZi Wang, Feng Xiao, QiZhou He, Ying Zhang, "Using a machine learning-based risk prediction model to analyze the coronary artery calcification score and predict coronary heart disease and riskassessment",doi:10.1016/j.compbiomed.2022.1 06297, ID- 106297, Volume 151,2022

[6] Hanna Kwiendacz, Agata M. Wijata, Jakub Nalepa, Julia Piaśnik, Justyna Kulpa, Mikołaj Herba, Sylwia Boczek, Kamil Kegler, Mirela Hendel, Krzysztof Irlik, Janusz Gumprecht, Gregory Y. H. Lip and Katarzyna Nabrdalik," Machine learning profles of cardiovascular risk in patients with diabetes mellitus: the Silesia DiabetesHeartProject",(2023)

[7] Jobeda Jamal Khanam, Simon Y. Foo, "A comparison of machine learning algorithms for diabetes prediction",ICT Express Volume 7, Issue 4,432-439 (2021).

[8] Liangjun Jiang A, Zhenhua Xia B, Ronghui Zhu C, Haimei Gong A, Jing Wang D, Juan Li E, Lei Wang, "Diabetes risk prediction model based on community follow-up data using machine learning" Volume 35,Article ID:102358,(2023).

[9] Chollette C. Olisah, Lyndon Smith, Melvyn Smith, "Diabetes mellitus prediction and diagnosis from a data preprocessing and machine learning perspective",Article ID:220 (2022) 106773.

[10] Mrs P.Renukadevi, M.Savitha, R.M.Soundharya, " Diabetics Prediction using Back Propagation Neural Network algorithm", Volume 12, Issue 2, Article ID: 2250- 1770 (2022).

[11] Nesreen Samer El_Jerjawi, and Samy S. Abu-Naser, "Diabetes Prediction Using Artificial Neural Network" (2018).

[12] Mythili Thirugnanam, Dr Praveen Kumar, S Vignesh Srivatsan, Nerlesh C R, "Improving the Prediction Rate of Diabetes Diagnosis using Fuzzy,Neural Network, Case based (FNC) approach, "Procedia Engineering ,Volume 38 ,1709 – 1718.

[13] Putri Sari Asiha , Yufis Azhara , Galih Wasis Wicaksonoa, Denar Regata Akbia , "Interpretable Machine Learning Model For Heart Disease Prediction", 8th International Conference on Computer Science and Computational Intelligence, doi:10.1016/j.procs.2023.10.544 (2023).

[14] Mandakini Priyadarshani Behera, Archana Sarangi, Debahuti Mishra , Shubhendu Kumar Sarangi, "A HybridMachine Learning algorithm for Heart and Liver Disease Prediction Using Modified Particle Swarm Optimization with Support Vector Machine" (2023).

[15] Senthilkumar Mohan, Chandrasegar Thirumalai, and Gautam Srivastava, "Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques" in IEEE special section on smart caching, communications, computing and cybersecurity for information-centric internet of things July 3, 2019, pp.81542-8155