Automated Weighbridge Load Monitoring and Rule-Based Overload Detection System

Kaveri V S
Department of CSE
Ramaiah Institute of Technology
Bangalore, India
kaveri17vs@gmail.com

Dr. Monica R Mundada
Department of CSE
Ramaiah Institute of Technology
Bangalore, India
monica@msrit.edu

Dr. Sowmya B J
Department of AI and DS
Ramaiah Institute of Technology
Bangalore, India
sowmyabj@msrit.edu

***Abstract*—Overloaded vehicles are detrimental to road infrastructure and create safety risks for the traveling commuters. The conventional weighbridges require manual inspections of the vehicles, which creates a lot of inconveniences and added costs to the infrastructure of the roads. The proposed system uses software-based automated weighbridge monitoring system for analyzing the load of the vehicles in real time using simulated data from weighbridge sensors. The software system can capture the load of the vehicles at multiple checkpoints, classify them as NORMAL or OVERLOADED using predefined rules, and store the data in a cloud database. The data is also visualized on a web-based dashboard in real time. The system demonstrates efficient processing and consistent rule-based classification. The system uses simulated data for the demonstration of its features. However, the software system can be integrated with load sensors from the vehicles to monitor the load of the vehicles in real time. This system can be used for the management of transportation systems and can prove to be cost-effective for these systems.**

***Index Terms*—Automated Weighbridge, Real-Time Monitoring, Rule-Based Classification, Cloud Database, Intelligent Transportation Systems**

I. Introduction

Overloading of heavy commercial vehicles is a major concern for transportation infrastructure - such deterioration caused by these loads impacts the life of roads, bridges and flyovers while presenting an increased risk of traffic accidents. Regulatory authorities have set limits on how heavy vehicles. They have put in place systems to weigh vehicles to make sure they are not too heavy. In a lot of places people are still checking manually heavy vehicles which takes a long time and can be susceptible to error [1].

Traditional systems for weighing vehicles require vehicles to stop at points so they can be weighed and then someone has to write down the weight and check it. These systems are better than looking at the vehicle to guess its weight but they can cause traffic jams and do not give us information about the vehicles right away [2]. New systems for weighing vehicles are being made that use controllers and communication devices to reduce the need for people to do things by hand. These systems can also use the internet to store information and let people look at it from anywhere and show what is happening in real time [3].

Recently new automated systems, for weighing vehicles are better. They are often complicated and expensive to set up and maintain, which makes it hard for some places to use them. A system that's not fully automatic or has manual entry that is writing down the information by hand is not good enough when we need to check a lot of vehicles or keep track of all the information in one place [4]. Methods can improve detection accuracy but the need for large datasets and computational power reduces their suitability for regulatory enforcement systems [5].

A rule-based approach to overload detection presents a practical alternative based on permissible load thresholds established by transport regulations. Such methods are computationally efficient and easy to interpret - all attributes that make them ideally suited for real-time monitoring applications. Such a software-centric approach provides a scalable and cost-effective solution. The use of machine learning and the search of cameras in vehicles has also been tried to find out what kind of vehicle it is and how heavy it is [6].

The paper presents the design and implementation of a software-based weighbridge automation system utilising simulated sensor data and rule-based logic for monitoring vehicle loads in real time. The system measures loads at checkpoints, ensures permissible limits are met, stores the data within a central database and displays the information via a web-based dashboard. The software solution described provides an excellent foundation upon which to integrate future physical load sensors, IoT devices and automated enforcement mechanisms.

II. Related Work

Many studies have explored how to automate weighbridge operations to avoid manual entry and increase vehicular load observations. Kumar et al. [1] created an IoT weighbridge that integrates with the cloud in which weights are automatically recorded and stored to facilitate data access. While Kumar et al. went through significant assessments to prove that access to such information is more reliable than manual entries, issues of reliance upon networks and scalability were concerning.

Patil et al. [2] created a system that recognizes weight at check-posts and automatically creates billing to avoid prolonged manual processing. However, this method requires the implementation of hardware devices uniquely which may add to installation and maintenance costs. Other embedded-system weighbridge solutions exist for paperless data collection.

For example, Sharma and Singh [3] created an automated weighbridge with load cells and microcontrollers to digitize readings. Jain and Rao [4] rely on a similar centralized database architecture for their automated billing solution at checkpoints. Machine learning and data-driven assessments for overload detection and transportation monitoring have also been studied. Joshi and Shah [5] assess the role of rules-based anomaly detection as a less intensive assessment than a learned model.

In a similar vein, Banerjee et al. [6] discuss intelligent check-post monitoring systems that use sensing devices in conjunction with cloud settings for real-time freight information. Other studies explore real-time monitoring with automated billing in transport check-posts.

While this improves measurement accuracy, the accuracy is highly sensitive to over-calibration or under-calibration in addition to conditions, as Reddy and Narayana [8] note. Hardware involvement proves less useful when scalability or remote monitoring is necessary.

Studies have begun to leverage IoT and cloud technologies for central observation and analysis. For example, creating a compliance monitoring system based on IoT technologies that integrates real-time and historical access to information.

Yet continuous network availability would be a concern, as would the infrastructural costs to implement a similar weighbridge. Edge- and fog-processing have been assessed for latency reduction and bandwidth reductions, as Mehta et al. [9] discuss, though such systems add levels of design complexity. While this model makes sense for certain applications due to its explainable basis and low processing, the machine learning models investigated in Fernandez et al. [14] require substantial data to be effective, which limits interpretability and regulation. Ultimately, the related work demonstrated that many studies exist to facilitate weighbridge automation using hardware, IoT, and cloud solutions.

But most place emphasis on collaborative hardware solutions or excessively data-driven models which fail to consider software reliability, transparency or simplified implementation. Thus, the present work applies a software-based, rule-based solution to assess simulated sensor data to successfully facilitate real-time vehicular load assessment and effective data logging and dashboard implementation while securing scalable and interpretable results.

III. Proposed Methodology

The proposed online weighbridge automation system follows a modular and sequential workflow to support real-time vehicle load monitoring and overload detection.

The methodology is designed to replicate the functional behavior of a real weighbridge system using software components, while allowing controlled testing through simulated sensor data. The overall system architecture consists of data generation, backend processing, rule-based verification, database storage, and dashboard visualization.

The proposed methodology ensures that records are stored securely in the database, and the results are displayed in real time through a web-based monitoring dashboard.

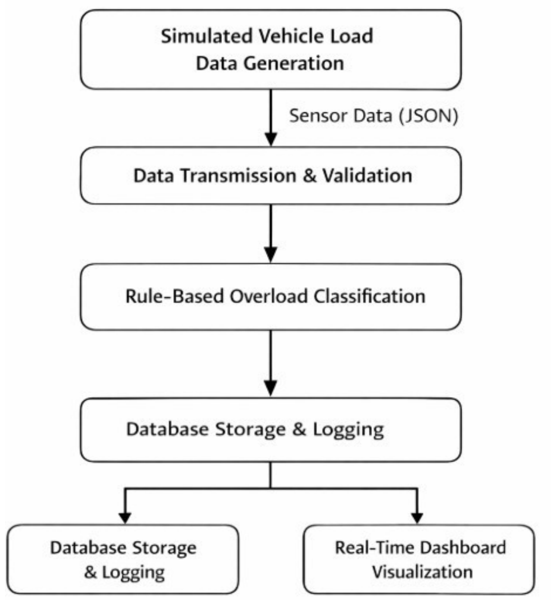

Fig. 1. Proposed Methodology

Fig. 1 shows us how the system works. The system starts by making vehicle load data. This data is sent in JSON format. Checked at the backend. The data that is validated is then checked by using rule-based logic to classify vehicles are normal or overloaded. The system uses vehicle load data to classify vehicles normal or overloaded vehicles.

A. Simulated Vehicle Load Data Generation

The system uses simulated sensors to estimate vehicle weight, allowing the monitoring process to be tested without relying on physical equipment. Each data entry includes details such as the vehicle identifier, checkpoint identifier, load in tons, and the corresponding timestamp. These records are generated at regular intervals to represent continuous vehicle movement across weighbridge checkpoints. This setup helps evaluate how the system processes and handles incoming data under realistic conditions while ensuring that the monitoring and classification functions operate correctly.

B. Data Transmission to Backend Server

The simulated vehicle load data is transmitted to the server using HTTP POST requests in JSON format, which enables structured and consistent communication between the data source and the backend system. When the server receives this data, it first validates the format to ensure correctness and then extracts the required fields for further processing. This stage reflects the communication layer of a real-world system, where sensing units and the processing server interact seamlessly. The vehicle load data plays a central role in this process, as it allows the server to interpret and respond to the ongoing conditions being monitored.

C. Rule-Based Load Verification

The system checks the vehicle load values after it gets all the data. It uses a set of rules to make sure the loads are okay. Each load is compared to the weight allowed by transport rules. If the load is too much the vehicle is considered OVERLOADED. If not it is considered NORMAL. This way the system is clear and consistent and does not take a lot of computer power.

Algorithm 1 Online Weighbridge Load Monitoring Using Rule-Based Logic

Require: Simulated vehicle load dataset D, permissible gross vehicle weight limit W_{max}
Ensure: Vehicle classification status (NORMAL / OVERLOADED), database records R, real-time dashboard updates D_{view}

```
Initialize backend server and database connection
Set permissible weight limit W_max
Initialize empty database table R ← ∅
for each dataset d ∈ D do
    Load vehicle records containing (Vehicle ID, Checkpoint ID, Load, Timestamp)
end for
for each vehicle record v ∈ D do
    Receive vehicle load data via HTTP POST request
    Validate data format and extract fields (V_i, C_i, W_i, T_i)
    if W_i > W_max then
        S_i ← OVERLOADED
    else
        S_i ← NORMAL
    end if
    Store (V_i, C_i, W_i, S_i, T_i) in database R
    Update real-time dashboard D_view with latest record
end for
Display vehicle load status and overload alerts on dashboard
```

As seen in Algorithm 1 the system gets the vehicle load data. Checks it to validate input parameters before it starts working. Then it compares the weight of each vehicle to the maximum allowed weight using the set of rules to see if it is overloaded or not. The vehicle load values are checked against the rules to determine if the vehicle is OVERLOADED or NORMAL.

The classified results are stored in the database and shown on the real-time dashboard. This process repeats to help with automated weighbridge monitoring and to make sure it is consistent. The classified results and the real-time dashboard are important for this to work.

D. Database Storage and Logging

The vehicle details, like the load value and overload status are stored in a database after they are classified. The database also stores the timestamp. Each record in the database has an index, which makes it easy to find and keep track of the vehicle details.

The database keeps a record of every time a vehicle moves across checkpoints. It helps with auditing and database also helps with monitoring the overload patterns of the vehicles over time. The vehicle details and the load value are very important and the database helps to keep track of them.

E. Real-Time Dashboard Visualization

A web-based dashboard is used to display the monitoring results in real time. It retrieves records from the database and presents key details such as vehicle ID, checkpoint ID, load value, overload status, and timestamp in a clear format. The dashboard updates automatically as new data arrives, allowing continuous tracking of vehicle activity across checkpoints. This real-time visualization helps authorities quickly identify overloaded vehicles and understand current conditions, making it easier to monitor operations and support timely decision-making.

F. Mathematical Formulation

1) Vehicle Load Representation: Each vehicle load record is represented as:

$$L_i = \{V_i, C_i, W_i, T_i\} \tag{1}$$

where:

- V_i is the vehicle ID
- C_i is the checkpoint ID
- W_i is the measured vehicle load (in tons)
- T_i is the timestamp of measurement

Each vehicle load record is like this: $L_i = \{V_i, C_i, W_i, T_i\}$. It has the Vehicle ID V_i is unique, to each vehicle. The Checkpoint ID C_i is where the measurement is taken. The vehicle load W_i is how much the vehicle weighs. The time T_i is when it was measured. This way of storing data helps us handle it easily.

2) Rule-Based Overload Classification: The overload decision is made by checking if the load's too high compared to the allowed limit. The classification is done like this:

$$S_i = \begin{cases} \text{OVERLOADED}, & \text{if } W_i > W_{max} \\ \text{NORMAL}, & \text{if } W_i \le W_{max} \end{cases} \tag{2}$$

where:

- W_{max} is the permissible gross vehicle weight limit
- S_i is the classification status of vehicle i

In this system overload detection is done by using rule-based logic. The vehicles load is compared to the allowed weight. If it is more the vehicle is OVERLOADED; otherwise it is NORMAL. This approach makes sure that decisions are clear and consistent. The classification result comes directly from predefined rules, about vehicle weights. It does not rely on models or probabilities.

3) Data Ingestion Processing Time: The delay in processing, which is called T_p is how long it takes for the system to handle a record of a vehicle load from when it's received to when it is shown.

$$T_p = T_{display} - T_{receive} \tag{3}$$

where:

- $T_{receive}$ is the time when the backend gets the data
-
- $T_{display}$ is the time when the data shows up on the dashboard

By subtracting the time the data gets to the backend server, which's $T_{receive}$ from the time the data appears on the dashboard, which is $T_{display}$. This measurement shows how well the system performs in time and to make sure that the data, about vehicle loads is handled and shown quickly with no delay. The system uses the processing delay T_p to evaluate its performance and the vehicle load data is processed using this metric to confirm that it is presented without any delay.

4) Overload Detection Accuracy: The accuracy of overload detection is measured by the number of classified vehicles, which is $N_{correct}$ compared to the total number of vehicles that are tested which is N_{total}. This is shown as a percentage. The accuracy of overload detection is about how correct the classification processs

$$\text{Accuracy} = \frac{N_{correct}}{N_{total}} \times 100 \tag{4}$$

where:

- The number of vehicles that are classified correctly is $N_{correct}$.
-
- The total number of vehicles that are tested is N_{total}.

The system makes decisions based on rules and fixed limits so this measure shows that the system always makes the decision, for all the vehicles that are tested. The system is consistent because it uses the rules every time to classify vehicles so it will make the right decision every time and this is confirmed by the accuracy of overload detection, which is calculated using $N_{correct}$ and N_{total}.

IV. Results and Discussion

The proposed online weighbridge automation system was evaluated through controlled experiments using software-based performance metrics. The analysis focused on how effectively the system processes real-time data, the accuracy of its rule-based overload detection, the reliability of database logging, and the responsiveness of the monitoring dashboard. Since the evaluation relied on generated input data rather than physical sensors, it primarily assessed the behavior and efficiency of the software components. As a result, hardware-related constraints and real-world deployment challenges were not considered in this evaluation.

A. Data Ingestion and Processing Time

The system was tested in a controlled environment by continuously sending vehicle data to the server for processing. Each incoming record was handled by the server, stored in the database, and then reflected on the dashboard. The dashboard displays key details such as vehicle ID, checkpoint ID, load value, overload status, and timestamp, enabling effective tracking of vehicle movement and overload conditions over time. It updates quickly as new data arrives, indicating that data ingestion and processing occur efficiently. Overall, the system manages vehicle load data smoothly and provides a reliable interface for real-time monitoring.

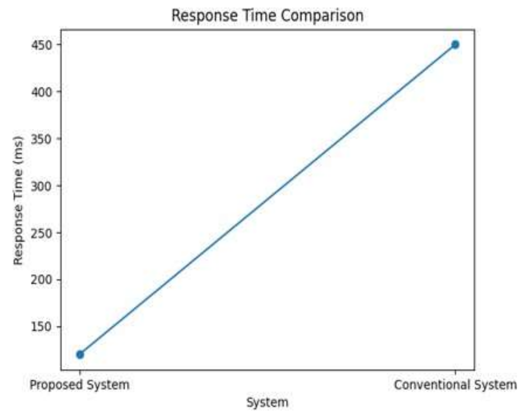

Fig. 2. Response Time Comparison

Fig. 2 shows how automated weighbridge system works compared to the manual system. The new system is much faster because it captures data automatically. This makes it quicker because the system gets the data on its own follows rule-based logic and then stores it. The automated data capture makes the system efficient and decisions are made faster and displayed in dashboard.

B. Rule-Based Classification Accuracy

The overload detection logic was evaluated by comparing vehicle load values against a predefined weight limit. Vehicles exceeding this limit were classified as OVERLOADED, while those within the limit were marked as NORMAL. Since the classification is based on a fixed threshold, the rule is simple and easy to apply. This approach produced consistent results across all test cases, with each vehicle correctly categorized according to its load value. Overall, the rule-based method performed as expected and ensured reliable classification.

C. Assessment of Database Storage Performance

The performance of a database storage system is about how fast and consistent it's when it reads writes and gets data. This happens when the system is working under conditions. After the records were into a database, the data shows that there are no records missing and no duplicates. The database kept the order in which things were added and it kept timestamps. This means it worked well for recording things in time and, for storing things for a long time. The database storage performance of the system was consistent.

D. Dashboard Update and Visualization

The monitoring dashboard was checked to see how it worked when new vehicle load data was put into the system. The monitoring dashboard updates to show the vehicle ID, the checkpoint ID, the load value and overload status. The monitoring dashboard updates to show all the new records. This shows that the backend and the frontend of the monitoring dashboard are synchronized. The monitoring dashboard and the vehicle load data are used for real-time monitoring.

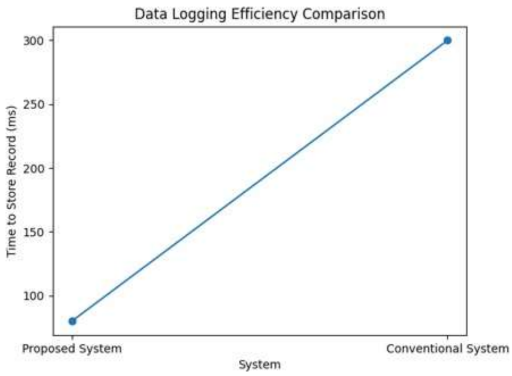

Fig. 3. Data Logging Efficiency

Fig. 3 shows us how long it takes to store the weight of vehicles in the database. The new system is faster because it does not need manual entry. Instead it uses sensors to get the data and sends it to the database. This way we can see what is happening away and it reduces the time. The new system records vehicle weight records shortens delays.

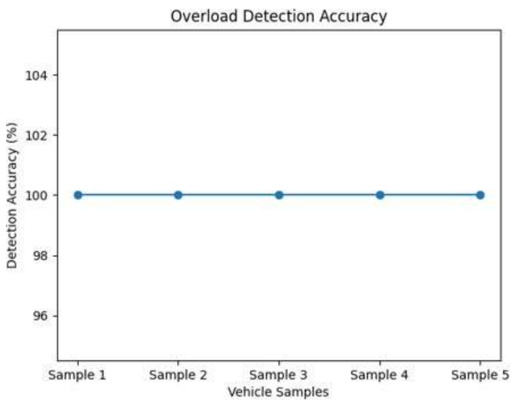

Fig. 4. Overload Detection Accuracy

Fig. 4 shows how well the proposed system can detect overload in different vehicle samples. The system uses predefined load limits to figure out if a vehicle is normal or overloaded. It does this by using a set of rules that help it decide whether each vehicle is NORMAL or OVERLOADED.

E. Performance Evaluation

The proposed online weighbridge automation system was done using sensor data that was simulted and sent to the backend server using APIs. Each record for a vehicle had a lot of information about the vehicle. This information included the vehicle ID, the checkpoint ID how much the vehicle was loaded the time it was weighed and whether the vehicle was overloaded or not. The online weighbridge automation system is what we are focusing on here. The vehicle ID and checkpoint ID are parts of the vehicle record, in the online weighbridge automation system.

TABLE I: Performance Comparison of Proposed and Conventional Systems

Parameter	Conventional System	Proposed System
Data Capture Method	Manual Entry	Automated
Response Time (ms)	~450	~120
Data Logging	Semi-automatic	Automatic Database Logging
Overload Detection	Human-dependent	Rule-based Logic

From Table I, it is observed that the proposed system significantly improves response time and reliability by eliminating manual intervention.

F. Performance Discussion

The response time comparison shows that the proposed system is really fast at processing vehicle data. It does this faster than the weigh-bridge systems. The dashboard visualization is also very useful. It lets authorities see away which vehicles are overloaded. This helps reduce the risk of damage to roads and bridges.

V. Conclusion

This paper is about designing a system to monitor vehicle loads in real time. The system uses rules to check if a vehicle is overloaded. It replaces data entry and verification with automated processes. Data is collected automatically, the system checks if a vehicle is overloaded. Information is stored in a database. Then the data is shown on a dashboard in time. Using limits for vehicle loads ensures fair and consistent decision-making. The results show that a software-based system is a solution for intelligent transportation systems. It is scalable and cost-effective.

The systems design works with physical sensors, IoT devices and automated enforcement systems. Some possible future improvements are: Automated billing systems, using RFID to identify vehicles, analyzing traffic trend and predicting overloads using machine learning. The system provides a base for scalable and intelligent weighbridge monitoring solutions. It can monitor vehicle loads in time and offers a scalable and cost-effective solution. The online weigh-bridge automation system uses rule-based logic. The system is suitable for integration, with physical load sensors and IoT devices.

References

[1] A. Kumar *et al.*, "Automation of weigh bridge systems using IoT and cloud-based data management," *IEEE Access*, 2021.
[2] S. Patil *et al.*, "Real-time load monitoring and billing at transportation checkpoints," *Procedia Computer Science*, 2020.
[3] N. Sharma and V. Singh, "Automated weigh bridge systems for smart transportation," in *Proc. Int. Conf. Smart Transportation Systems*, 2022.
[4] R. K. Jain and P. T. Rao, "Centralized database architecture for automated billing," *International Journal of Computer Applications*, 2021.
[5] M. Joshi and D. Shah, "Rule-based anomaly detection for load monitoring," *IEEE Access*, 2021.
[6] A. Banerjee, S. Dutta, and P. Mukherjee, "IoT-enabled intelligent check post monitoring for freight transportation," in *Proc. IEEE Int. Conf. Smart Infrastructure*, pp. 210–215, 2021.
[7] K. Ramesh and S. Balakrishnan, "Automated billing and auditing framework for weigh bridge operations," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 6, pp. 402–408, 2020.
[8] P. S. Reddy and K. L. Narayana, "Smart transportation systems using sensor-based load measurement," in *Proc. Int. Conf. Internet of Things and Applications*, pp. 95–100, 2019.
[9] S. Mehta, R. Patel, and H. Trivedi, "Edge-assisted processing for real-time transportation monitoring," in *Proc. IEEE Int. Conf. Distributed Computing Systems*, pp. 310–317, 2020.
[10] A. Kumar and M. Singh, "Design of scalable architectures for smart flyover monitoring systems," *International Journal of Computer Applications*, vol. 176, no. 34, pp. 1–6, 2021.
[11] R. S. Iyer and P. Mahajan, "Data synchronization techniques for distributed transportation checkpoints," *Procedia Engineering*, vol. 187, pp. 532–539, 2019.
[12] T. G. Rao and S. K. Mishra, "Centralized monitoring and analytics for intelligent transportation infrastructure," in *Proc. IEEE TENCON*, pp. 1254–1259, 2020.
[13] J. Fernandez, L. Gomez, and M. Ruiz, "Automated freight monitoring using IoT-based sensing and cloud platforms," *Sensors*, vol. 20, no. 18, 2020.
[14] S. Kaur and R. K. Singh, "Secure data management in automated transportation systems," *Journal of Network and Computer Applications*, vol. 168, 2020.
[15] V. N. Prasad and A. Chandra, "Real-time alert generation for smart transportation checkpoints," in *Proc. IEEE Int. Conf. Intelligent Systems*, pp. 455–460, 2021.