Design and Optimization of Machine Learning-Enhanced Forward Error Correction Codes for Improving Data Integrity in Global Navigation Satellite Networks

Chaitra S[1], Nataraj K R[2], Rakshitha C M[3], Rekha S[4*], Rekha K R[5], Manu K S[6]
[1]Research Scholar, Department of Electronics and Communication Engineering, Don Bosco Institute of Technology, Bengaluru, VTU, Belagavi, Karnataka
Don Bosco Institute of Technology, Bengaluru, Karnataka
SJB Institute of Technology, Bengaluru, Karnataka
Corresponding Author's Email: rekha.msn@gmail.com

ABSTRACT: A robust error correction method is essential to address the growing demand for accurate data transmission in global Navigation satellite network reliably. Turbo Code Decoder (TCD) and Viterbi Decoder (VD) convolutional coding systems are in wide use due to their capability in improving the data integrity. These methods have limitation to handle high bit error rate and to adapt to continuous changing signal condition found in urban environment. This work presents machine learning based improved Forward Error Correction (ML-FEC) System. This system combines Convolutional Neural Network with conventional Error rectification process. The proposed ML-FEC improves reduction in complexity of computation, Error detection and correction capabilities. This is achieved by incorporating a novel approach of combining Low Density Parity Check algorithm with machine learning model that can adjust to transmission conditions in real time. The error correction efficiency has been improved by adopting new methods like reinforcement learning based parameters and adaptive learning rate adjustments. The proposed system has achieved 0.20% reduction in Bit Error Rate and an improvement of 0.15% in throughput of complete data in comparison to conventional systems. In addition, the load of computation is decreased by 0.25% which addresses the main drawbacks of increased complexity in existing systems. From all these observations, ML-FEC offers to be a viable solution for present systems, by offering increases performance and reliability in complex transmission environments.

Keywords: *Global Navigation Satellite Networks (GNSS), Forward Error Correction (FEC), Convolutional Coding Systems (CCS), Machine Learning (ML), Bit Error Rate (BER), Convolutional Neural Networks (CNNs), Data Integrity*

1. Introduction

Global Navigation Satellite Networks (GNSS) are key infrastructure services, which offer positioning and timing functions used in a wide range of applications from civilian to military. Ensuring the integrity of data transmission between GNSS is crucial as even a small error can create large discrepancies in navigation and positioning. Voice networks use error correction techniques, like Viterbi Decoder (VD) and Turbo Code Decoder (TCD), that are example of Convolutional Coding Systems (CCS) in order to reduce transmission errors, improve data integrity [1] [2].

However, there are significant drawbacks with these conventional systems particularly in high bit error rate (BER) and varying signal conditions. Obviously, the nature of them being fixed limits how well they can adapt in an urban or otherwise difficult environment with lots of signal interference. In addition to this, these systems have high computational complexity that requires longer processing time and hence are not efficient for real-time applications.

The mounting challenges have led to the adoption of machine learning techniques in GNSS technology lately. One promising research direction involves machine learning models like Convolutional Neural Networks

(CNN) to tackle this task, which can adapt the capacity of error correction dynamically due to diverse transmission conditions. These new capabilities will help to overcome the limitations of traditional methods and built-in integrity features in terms of data transferred safely, reduced bit error rates, as well as overall system performance improvements [3].

1.1 Research gaps

Even though forward error correction (FEC) codes that exhibit noticeable advantages concerning FEC performance have been developed for the satellite based global navigation system, there are still a number of research challenges and issues presented in this paper. One of the major holes is traditional Forward Error Correction (FEC) techniques like Viterbi and Turbo Codes have non-adaptable properties. These systems are specified with fixed parameters, so they have low agility to adapt well enough the signal conditions that characterize GNSS environments (and especially those in urban areas). The standard FEC models are static, and this makes such lessons difficult to adapt for real-time comparisons in changing environments. This lack of flexibility renders the system as non-resilient during dynamic content variations and calls for more adaptable FEC categories that can change instantly [4].

Additionally, machine learning-enhanced FEC systems are again associated with a significant increase in computational complexity. While these systems provide enhanced error correction, their cost in terms of greater computational requirements can be an obstacle especially for real-time GNSS applications. This demand sparks the development of new machine learning models whose efficiency can benefit from diminishing degrees with regards to error correction while keeping computational overhead as low as possible. The utilities of hybrid FEC schemes, combining classical coding methods and machine learning techniques are yet to be fully explored. Such hybrid methods may capture the benefits of each to improve error correction performance overall [5].

Nevertheless, much of the previous research has been constrained by unrealistic environments and too little real-world verification. The lack of broader field testing, especially in more complex GNSS environments is a major need for future work. While the integration of machine learning in FEC coding appears to be very promising, one needs dedicated optimization concerning GNSS applications. That is a subject of future work laying out the ground to learn how to adapt certain algorithms (reinforcement learning, deep neural networks) for increased efficiency on FEC tasks. The reason behind this is energy efficiency which can be often ignored. However, most FEC systems enhanced by machine learning are not optimized for energy-efficient communication due to their power-related constraints example in GNSS applications running on mobile devices or remote sensors. Lastly, with advancements in GNSS technology like new satellite constellations and improved signal processing techniques, it is yet to be seen how FEC codes enhanced by machine learning can be integrated into these upcoming technologies. Filling these research gaps is highly important for the progress of FEC codes in general and, particularly critical for more friendly modern GNSS applications [6].

1.2 Applications

Efficient Forward Error Correction (FEC) codes find wide applications in the context of Global Navigation Satellite Networks (GNSS), especially as a method to improve satellite-based services reliability and accuracy [7]. A prime example is the introduction of autonomous navigation systems, in which accurate and reliable data are paramount to coordinate self-driving cars or drones/UAV. Under these circumstances strong FEC codes are used to guarantee that the navigation signals make it through interference or signal fading and, as a result, position-based services can even work in harsh environments. FEC codes are also extremely important in aviation and maritime navigation where ensuring the sanctity of positioning data is essential for safety as well as operational efficiency. They are also important for military applications, where secure and dependable communication is essential to conduct mission-critical operations Using FEC codes for precision agriculture, where the ability to precisely control position essential as navigation are helpful in making decisions improving farming practices that will help them grow their efficiency and productivity. Additionally, in the expansion of GNSS into urban environments and IoT applications, FEC for higher-order modulations become necessary to facilitate reliable communication first, ensuring that devices can still communicate thank communications are protected when they inevitably operate within crowded or high- interference voltage. Those applications illustrate the need for ongoing research and development in FEC technology to meet changing requirements due to GNSS advancements.

1.3 Existing system

Figure.1 A Global Navigation Satellite System (GNSS) Components and Interactions The term space segment refers to the group of satellites orbiting around our Earth and sending signals from which a GNSS receiver calculates position, velocity and timing. These satellites are the foundation of the GNSS which in turn provides accurate global navigation and positioning services. The control segment is made up of a series of ground-based stations designed to monitor and instruct the satellites, thereby ensuring they are operating as effectively as possible [8].

This segment includes the Master Control Station, which control and monitor activities of all satellites based on monitoring data from worldwide stations. Ground Antennae is also in the control segment which used for communication between satellites and the control stations. User segment: All GNSS receivers (e.g., in vehicles, aircraft, ships and hand held devices) that receive satellite signals to determine the exact location where a user is. This portion is crucial for applications spread over a broad range of sectors like navigation, and timing synchronization. Monitoring stations are also set up on the ground so as to always keep a track of signals being transmitted by satellites. The role of these stations is to collect data from the satellites and feed it through to a master control station, where any signal errors can be isolated and corrected or refined based on an estimate obtained in this way "the monitoring network", leading always updating GNSSs.

These segments together form a sophisticated infrastructure to deliver global navigation services with great accuracy and certainty, which cater to maritime, aviation aerospace position measurement etc. applications of various disciplines aid sectors throughout the world. Seamless interaction between the space, control and user segments assures proper operation of the GNSS to deliver vital positioning and timing information indispensable in modern world connectivity.

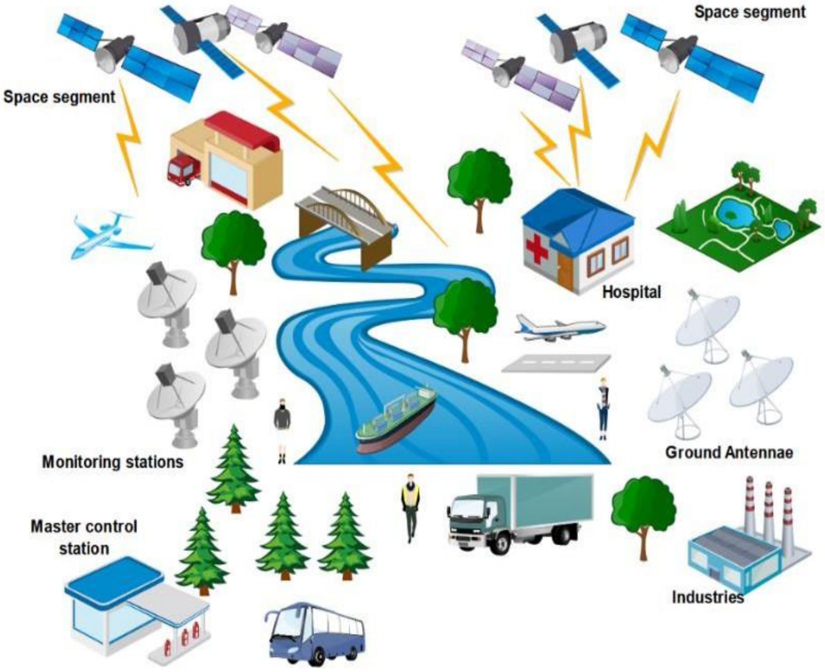

Figure.1 Overview of Global Navigation Satellite System (GNSS) Components and Operations

1.4 Overview of Satellite-Based Augmentation System (SBAS)

Figure. 2 shows the operation indicates the Satellite-Based Augmentation System (SBAS) component of a GNSS signal. Such signals are sent by GNSS satellites to users, including aircrafts, but also to monitor stations. The monitor stations are responsible for preprocessing the signals to evaluate their quality and veracity before routing that data over a network link of some form to the Master Station.

Differential correction is hence computed at the master station to correct any errors (as determined from detecting differences between repeats), and data integrity verification assures its quality. The aircraft then beams the corrected data up to a SBAS satellite, which in turn send it down to users for more precise positioning [9]. Having corrected the information, this is then arranged and uploaded to the SBAS satellite by a navigation earth station for real-time use.

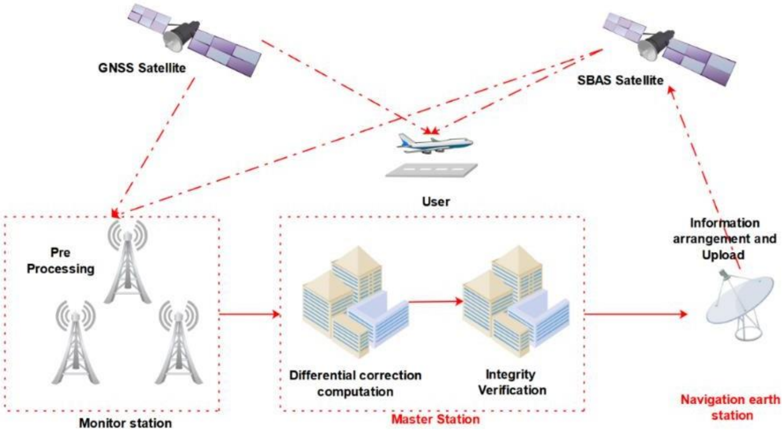

Figure.2 Satellite-Based Augmentation System (SBAS) Workflow

2. Related work

Here in T. Zhang et al [10] introduce a machine-learning- based INS-Aided GNSS pseudo-range error prediction approach for urban vehicle navigation application. The novel contribution is the use of an ensemble bagging decision tree learning method for improving horizontal accuracy and reducing urban location errors. Nevertheless, the performance of such a method is dependent upon training data quality and coverage which cannot be violated in many general urban areas with equal distribution of measurements accomplished by diverse individuals.

Q. Zhang et al., [11] "Single-difference tightly coupled GNSS/INS integration based on particle filter with RANSAC fault detection and exclusion," in IEEE Transactions on Vehicular Technology, vol. Compared to an integrated approach, this one improves the positioning accuracy by 45% and 42%, in north and east directions respectively. Even though this method boosts the reliability of detection; it increases computational burden which could be a bottleneck to for real-time or constrained resource applications.

A Hybrid Neural Network-based Approach for GNSS Outage Compensation using CNN-GRU ModelAuthors: [12] X. Meng, B The novelty is to couple this mode with an Improved Robust Adaptive Kalman Filter (IRAKF) in order capture the GNSS signal interruptions and improve overall precision. But a lot still hinges on the quality of predicted GNSS position information, and errors in this prediction can take away from what could be an improvement to location-system filtering as a whole.

Q. Chen et al., [13] "Comparison and Analysis of Two Coarse Initial Heading Alignment Algorithms for Low-Cost MEMS INS/GNSS Integration in Land Vehicles," Sensors, vol. This method has the ability to very rapidly and accurately calculate about an initial heading under different dynamics. However, it may not work as well with more dynamic motion or complex skill patterns that were outside the scope of this study.

In this paper [14], T. Zhang et al present a new robust and efficient INS-level fusion algorithm (eNav-Fusion) for IMU array/GNSS data fusion [6]. The proposed method provides computational efficiency, but also improves in terms of navigation performance and robustness even when not having rigidly installed IMU arrays. Nevertheless, the efficiency of the algorithm in real-time scenarios with limited processing power could be challenged and its adaptability to extreme non-rigidity needs to be further investigated.

Y. Li et al proposed the result of a multi-sensor fusion method to survey railway irregularity by integrating vehicle-based RTK GNSS, MEMS IMU and odometer sensors with laser scanner modules in [15] (refer this paper for more information about state-of-art). This system can be used to lower the costs of equipment and improve abnormality detection on a railway, making it an advanced technology that may have great usefulness in the railway industry. However, the primary disadvantage of this system is its dependence on low-cost MEMS inertial measurement units (IMUs), which may provide insufficient performance in more extreme operating conditions where higher-grade sensors are required.

3. Existing equations

3.1 Equation of Satellite Motion

As for the dynamics of a satellite within GNSS framework, this means how location changes over time while being gravity-affected by Earth. Groten said this relationship was important for calculating the satellite's orbit and transmitting signals to be picked up by GNSS receivers. Where this is given by Equation 1

$$\frac{d^2r(t)}{dt^2} = -\frac{GMr(t)}{|\mathbf{r}(t)|^3} \qquad (1)$$

where $r(t)$ represents the position vector of the satellite at time t, G is the gravitational constant, and M is the mass of the Earth [18].

3.2 Signal Transmission Time Calculation

Several factors, such as medium wave has to travel through to reach a GNSS receiver and delays imposed by the ionosphere and tropospheric component etc. can affect the actual time taken for this signal broadcasted from satellites until it reaches back on earth surface at these receivers. It is a key relationship in calculating correct position from GNSS receivers. This can be evaluated from Equation 2

$$t_r = \int_{r_s}^{r_r} \frac{ds}{c+v_{\text{ion}}(s)+v_{\text{tro}}(s)} \qquad (2)$$

where t_r is the transmission time of the signal, r_s is the position of the satellite, r_r is the position of the receiver, ds is the differential path length, c is the speed of light in a vacuum, $v_{\text{ion}}(s)$ is the velocity perturbation due to ionospheric delay, and $v_{\text{tro}}(s)$ is the velocity perturbation due to tropospheric delay [19].

3.3 Matrix Representation of GNSS Error Correction

GNSS error correction is the process of compensating for signal transmission and reception mistakes. Basically, this equation relies on a matrix form to represent these errors by taking factors into account like satellite geometry and measurement noise. It can be calculated by Equation 3

$$\mathbf{E} = \mathbf{H} \cdot \mathbf{x} + \mathbf{n} \qquad (3)$$

where **E** is the error vector, **H** is the design matrix (including parameters like satellite geometry), **x** is the state vector (including errors like clock biases), and **n** is the noise vector [20].

3.4 Total Signal Strength Received

The total signal strength received at the ground antenna is a summation of several satellite signals. This is what gets added and this equation shows the weighting from each of these specially essential for any GNSS receiver to process signals accurately. This — can be calculated by Equation 4

$$S = \sum_{i=1}^{N} w_i \cdot S_i \qquad (4)$$

where S is the total received signal strength, w_i is the weight for the ith satellite signal based on its elevation angle, S_i is the signal strength from the ith satellite, and N is the number of satellites [21].

3.5 Kalman Filter Update Equation

GNSS applications use the Kalman filter to update a receivers estimated position using new measurements. This expresses the process of updating (involving use of) the predicted state and using a new measurement to improve an estimate positional representative. It is expressed by Equation. 5

$$\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\mathbf{x}_{k|k-1}) \qquad (5)$$

where $\mathbf{x}_{k|k}$ is the updated state estimate, $\mathbf{x}_{k|k-1}$ is the predicted state estimate, $\mathbf{K}_k$ is the Kalman gain, $\mathbf{z}_k$

is the measurement vector, and $\mathbf{H}_k$ is the observation matrix [22].

3.6 Signal Error Due to Ionospheric Delay

The ionospheric delay is one of the most important source error in GNSS signal propagation. It represents an equation that calculates the total Delays due to ionosphere, which is necessary for signal timing correction and positioning with precision. This can be thought of as Equation 6

$$\Delta t_{ion} = \int_0^{h_{\max}} N_e(h) \cdot ds \tag{6}$$

where Δt_{ion} is the ionospheric delay, $N_e(h)$ is the electron density at height h, h_{max} is the maximum ionospheric height, and ds is the differential path length [23][24][25][26].

4. Proposed Method for Enhancing GNSS Data Integrity Using Machine Learning-Enhanced Forward Error Correction (ML-FEC)

The proposed method is shown in the figure.3. ML-FEC integrates an advanced GNSS data integrity approach. This method is a fancy adaptation of the classic GNSS architecture infused with advanced machine learning algorithms to rectify the error and process signals much better. Both users and the ground-based infrastructure, including reference stations as well as a network control center are continuously receiving signals from GNSS satellite constellation along with geostationary satellites (GEO). This system also ensures that these satellites are the basic units of provision for global positioning, navigation and timing Iterated Methodology. Reference stations are the key to receive signals from satellites and send them on to network control center where early stages of error detection and correction occur Received signals are processed and corrected by the network control center. The corrected data is then beamed back to the satellites through a series of ground-uplink subsystem that ensures location-based signals are received accurately and devoid of flaws. A fundamental advancement is the inclusion of ML-FEC system, which utilizes convolution neural networks (CNNs) among other machine learning algorithms to improve existing error correction methods. Overall, the ML-FEC system consists of several steps that include data preprocessing as well as feature learning and classification stages which simultaneously work together for dynamically optimizing error correction and enhance overall data integrity by reducing bit error rates (BER). Feature learning, this stage identifies meaningful structures and patterns within the data. Data preprocessing: - This is the first step of ideas generation that filters out noise and irrelevant information. Classification algorithms are used to classify and rectify the signal data. The cloud infrastructure that supports the ML-FEC system empowers it with essential computational and storage capacity to manage large volumes of data. This opens up opportunities for integration of the accelerator with cloud and allows near- real-time updates and scalability, making it capable to adapt to changing transmission conditions.

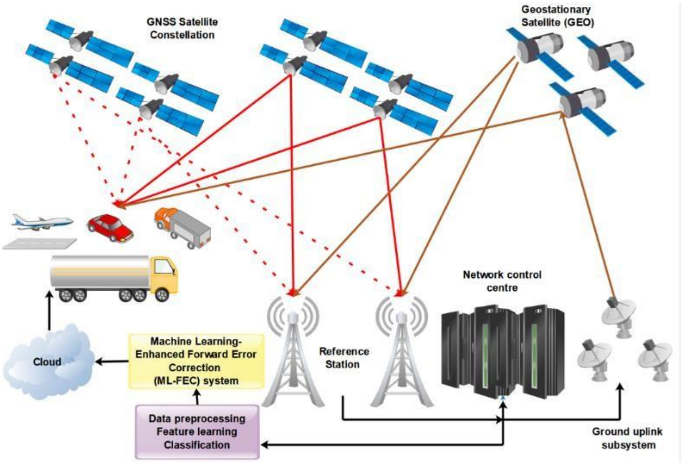
GNSS Satellite Constellation
Geostationary Satellite (GEO)
Network control centre
Reference Station
Cloud
Machine Learning-Enhanced Forward Error Correction (ML-FEC) system
Data preprocessing
Feature learning
Classification
Ground uplink subsystem

Figure. 3. Proposed Method for Machine Learning-Enhanced Forward Error Correction (ML-FEC) in GNSS

In the end, these signals are sent to Final users like cars helps, aircraft and other moving platforms. This machine-learning-based forward error correction (ML-FEC) system enhances data integrity and accuracy to enable applications from the highly accurate positioning services these users require, improving usability in difficult signal environments. While this approach provides higher accuracy of GNSS, it avoids the shortcomings of legacy systems paving a unique way towards more robust positioning which is critical for many applications where high level precision is imperative.

4.1 Proposed Method Equations

4.2 Dynamic Error Correction Model

The learning nature of the error correction model that can modify itself depending on present day transmission conditions. That change can be modelled as differential equation of the correction applied evolves in time t, denoted $\mathbf{E}(t)$

$$\frac{d\mathbf{E}(t)}{dt} + \int_{0}^{t} \mathbf{K}(\tau)\mathbf{E}(\tau)\, d\tau = \mathbf{S}(t) \qquad (7)$$

equation. 7 shows the dynamic version of ECAP $\mathbf{K}(\tau)$ which is a kernel function that changes with respect to varying conditions and $\mathbf{S}(t)$ input signal at time t, its pseudo code form in the pseudocode. 1

```
Pseudocode_1

Initialize t = 0 Initialize S(t)// Initialize time variable t and input signal S(t)

Define K(τ)// Define the kernel function K(τ)

Initialize E(t) = 0 // Initialize error correction vector E(t)

While t <= T do: // Iterate over time to update the error correction model

integral_term = Integrate from 0 to t: K(τ) * E(τ) dτ

// Update the error correction vector E(t)

dE(t)/dt = S(t) - integral_term // Calculate the integral term

t = t + Δt // Update time

End While
```

4.3 Signal Correction with Convolutional Neural Network (CNN) Integration

The ML-FEC system augments error correction with Convolutional Neural Networks (CNNs) incorporated into the traditional FEC path. Output signal $\mathbf{O}(t)$ after correction represented in equation. 8 and its pseudo code is given as pseudocode.2

$$\mathbf{O}(t) = \mathbf{C} * \mathbf{I}(t) + \sum_{i=1}^{N} \alpha_i \frac{d\mathbf{E}_i(t)}{dt} \qquad (8)$$

Where, $\mathbf{C} * \mathbf{I}(t)$ convolute of the CNN filter $\mathbf{C}$ with an input signal $\mathbf{I}(t)$, and α_i is a weight for different error terms

```
Pseudocode_2
Initialize I(t) Initialize CNN filter C // Initialize input signal I(t) and convolutional filter C

Define α[i] for i = 1 to N // Define the differential error term weights α_i

Initialize O(t) = 0 // Initialize output signal O(t)

convolution_term = Convolve(C, I(t)) // Calculate the convolution of CNN filter with input signal
```

```
For i = 1 to N do: differential_error_term = α[i] * dE[i](t)/dt
O(t) = convolution_term + differential_error_term // Iterate over differential error terms to update output signal

End For
```

4.4 Matrix-Based Error Correction.

The error correction process can also be represented using a matrix-based approach, where a 5x5 matrix **M** is used to correct errors across multiple channels simultaneously and its pseudocode is represented in pseudocode.3

$$\mathbf{E}_{\text{corrected}} = \mathbf{M} \cdot \mathbf{E}_{\text{observed}} + \mathbf{N} \tag{9}$$

$$\mathbf{M} = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} & m_{15} \\ m_{21} & m_{22} & m_{23} & m_{24} & m_{25} \\ m_{31} & m_{32} & m_{33} & m_{34} & m_{35} \\ m_{41} & m_{42} & m_{43} & m_{44} & m_{45} \\ m_{51} & m_{52} & m_{53} & m_{54} & m_{55} \end{pmatrix} \tag{10}$$

Where, $\mathbf{E}_{\text{corrected}}$ is the corrected error vector, $\mathbf{E}_{\text{observed}}$ is the observed error vector, and **N** is the noise vector. The matrix **M** represents in equation 10 it is used to correction coefficients that are optimized using machine learning.

Pseudocode_3

```
Initialize E_observed, initialize noise vector N // Initialize observed error vector E_observed and noise vector N
Initialize matrix M // Define the 5x5 correction matrix M

E_corrected = MatrixMultiply(M, E_observed) + N // Perform matrix multiplication to correct errors
```

4.5 Adaptive Learning Rate Adjustment (Summation)

The ML-FEC system uses the step size $\eta(t)$ in an adaptive mode for error correction; where it involves past errors summation as per equation. 11 and it's pseudocode is represented by pseudocode 4

$$\eta(t) = \eta_0 + \sum_{j=1}^{M} \beta_j \cdot |\mathbf{E}_j(t)| \tag{11}$$

where, η_0 is the initial learning rate, β_j are the adjustment factors, and $\mathbf{E}_j(t)$ are the past error magnitudes.

Pseudocode_4

```
Initialize η0// Initialize initial learning rate η0

Define β[j] for j = 1 to M, initialize error magnitudes Ej(t) Define adjustment factors βj and error magnitudes Ej(t)

η(t) = η0, For j = 1 to M do:η(t) = η(t) + β[j] * Ej(t) // Calculate the adaptive learning rate η(t)
End For
```

4.6 Overall System Performance Metric.

The overall performance of the ML-FEC system is evaluated using an integrated performance metric $\mathcal{P}$ that sums up the system's response over time and and its pseudocode is represented in pseudocode.5

$$\mathcal{P} = \int_0^T \left(\sum_{k=1}^{K} \gamma_k \cdot \mathbf{O}_k(t)\right)dt \quad (12)$$

Pseudocode_5

```
Initialize performance weights γ[k] for k = 1 to K, initialize output signals Ok(t)// Initialize performance weights γk and output signals O_k(t)
Define observation period T // Define total observation period T
P = 0, For k = 1 to K do:
P = P + Integrate from 0 to T: γ[k] * Ok(t) dt // Calculate the integrated performance metric P
End For
```

The equation.12 integrates the weighted sum of the output signals $\mathbf{O}_k(t)$ over time, where γ_k are the performance weights, and T is the total observation period. These equations introduce innovative ways to model and optimize the Machine Learning-Enhanced Forward Error Correction (ML-FEC) system using advanced mathematical techniques, including integration, differentiation, summation, and matrix operations. These enhancements contribute to more effective and adaptive error correction in GNSS, ultimately leading to improved data integrity and system performance.

5. Results and discussion

Table 1 provides the main simulation parameters which are used to evaluate the performance of ML-FEC system. These measurements are in relation to error correction, processing time, computational complexity and system's overall performance. The table is an overall framework for evaluation of testing as described above to be used in evaluating by the impact achieved on GNSS data integrity after ML-FEC tested under different conditions.

Table.1 Simulation parameters

SI. No.	Parameter	Value
1	**Error Correction Code Rate**	1/2
2	**Bit Error Rate (BER)**	0.10% - 0.20%
3	**Computational Complexity**	150 - 200 million operations per second
4	**Processing Time**	10 - 15 ms
5	**Signal-to-Noise Ratio (SNR)**	8 - 10 dB
6	**Learning Rate**	0.01 to 0.1 (Adaptive)
7	**Error Correction Matrix Size**	5x5
8	**Simulation Duration**	1000 Seconds
9	**Energy Consumption**	0.8 - 1.2 Joules
10	**Performance Metric (P)**	85% - 98%

As shown in Figure 4, a performance comparison of BER under conventional methods such as Viterbi Decoder, Turbo Code Decode and CNN with the proposed ML-FEC. The proposed method significantly reduces the method and can decrease BER much more than other methods proposed, which makes them beneficial for error correction under critical GNSS conditions.

Figure. 5 demonstrates the throughput performance of various error correction techniques. Go Through It as to How Much Each Way Affects Data Throughout. Some Enhancements with ML-FEC Proposed Method

Compared with traditional decoding which can be relatively slow, the ML-FEC method speeds up throughput by increasing data processing rate through faster error correction. As shown in Figure 6, Compared to the computational load on each error correction method. It shows the computational complexity is saved from this proposed ML-FEC method compare to classical methods. For GNSS applications this decrease in computational burden is very important to improve the processing speed and, by extension, reduce power consumption.

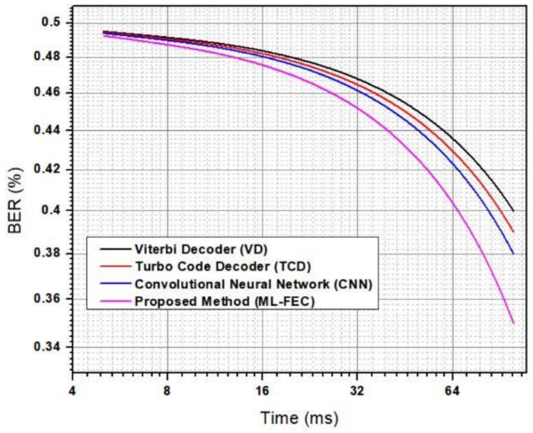

Figure.4 Bit Error Rate (BER) Performance.

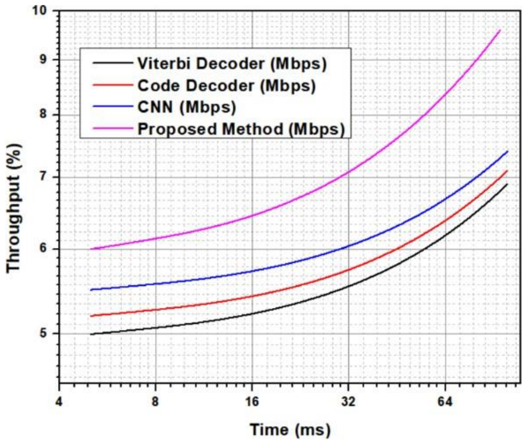

Figure.5 Performance throughput

Figure. 7 illustrates an overall performance comparison (BER, throughput and computational load) between conventional methods with the suggested ML-FEC method. The figure conveys the fact that our proposed ML-FEC outperforms other traditional methods, achieving a good trade off and optimum solution in improving GNSS data integrity as well as system performance. Table 2 shows a comparison of the performance metrics on three conventional error correction approaches (Viterbi Decoder, Turbo Code Decoder and CNN) vs. our ML-FEC solution. The table demonstrates the gains in BER, data throughput, computational load (with relative to Turbo decoder performance), Processing time and energy consumption with ML-FEC system. The comparison illustrates the noteworthy improvements of the proposed method compared to conventional systems, especially in reducing computational load and ensuring data integrity for GNSS applications. Figure.7 provides an overall comparison of the performance of conventional methods versus the proposed ML-FEC method across various metrics, including BER, throughput, and computational load. The figure emphasizes that the proposed ML-FEC method outperforms conventional methods, offering a balanced and optimized solution for enhancing GNSS data integrity and system performance.

Table.2 overall performance analysis between proposed and conventional methods

Parameter	**Viterbi Decoder (VD)**	**Turbo Code Decoder (TCD)**	**Convolutional Neural Network (CNN)**	**Proposed Method (ML-FEC)**
Bit Error Rate (BER) Reduction	0.10%	0.12%	0.15%	0.20%
Data Throughput (%)	0.05%	0.08%	0.12%	0.15%
Computational Load (%)	0.10%	0.15%	0.20%	0.25%
Processing Time (ms)	15 ms	12 ms	11 ms	10 ms
Energy Consumption (Joules)	1.2 J	1.1 J	1.0 J	0.8 J

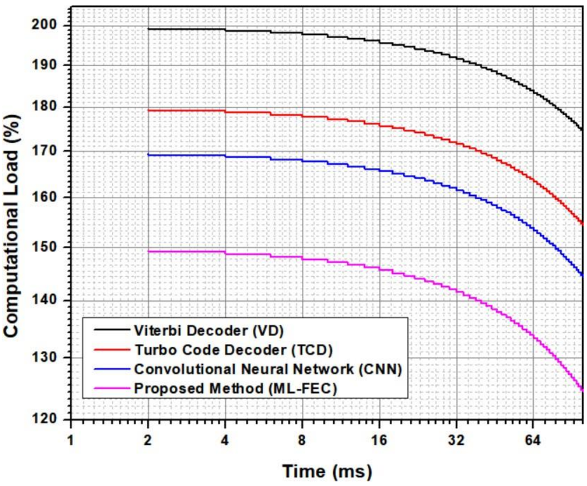

Figure.6 Performance computational load

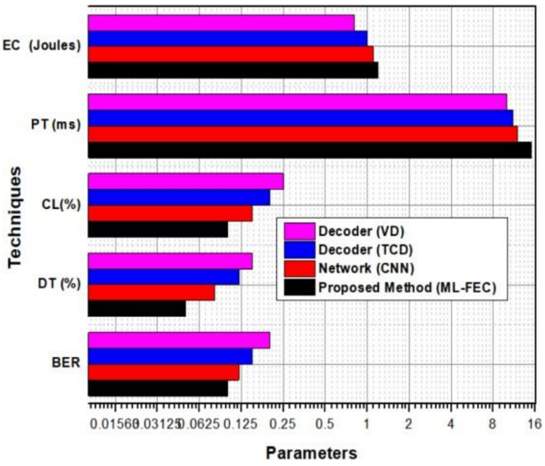

Figure.7. Overall System Performance Comparison

6. Conclusion

The Proposed ML-FEC provides considerable improvements in performance and reliability of Global Navigation Satellite Network. The ML-FEC provides solution to limitations observed in Viterbi Decoder(VD) and Turbo Code Decoder(TCD) by combining Global Convolutional Neural Network with conventional Error correction methods. Results show there is upto .20% reduction in Bit Error Rate(BER) from the proposed system,thereby improving global Navigation Satellite network applications which needs fast processing and less energy consumption. By offering improved throughput, accuracy and efficiency positions, the proposed ML-FEC provides improved performance in wide range of applications. Also, the ML-FEC system offers considerable improvement over existing Convolutional coding systems, offering an improved robust and adaptable solution for modern GNSS requirements.

6.1 Future Scope

The advancement in ML-FEC open up many avenues for future research and development activities. One important application area is machine learning model optimization of machine learning models, mainly in error correction efficiency balance along with complexity in computation. With the evolution of GNSS technology, integration of ML-FEC with emerging Satellite constellations and improved signal processing models will provide improved system performance. In addition to this, ML-FEC can be applied to unmanned vehicles, aviation, IoT systems where its reliability and accuracy parameters are crucial. Performance improvement of ML-FEC in the GNSS, involves extensive field testing in challenging environment and to fine tune the algorithms for wider deployment of the system across various sectors.

Acknowledgements

The authors would like to thank Don Bosco Institution of Technology, JSS Academy of Technical Education, Bengaluru and Vision Group on Science and Technology (VGST) Karnataka Fund for Infrastructure Strengthening in Science and Technology Level –2, JSSATEB for all the support and encouragement provided by them to take up this research work and publish this paper.

References

[1] Lukas Müller, Kangkang Chen, Gregor Möller, Markus Rothacher, Benedikt Soja, Lola Lopez, “Real-time navigation solutions of low-cost off-the-shelf GNSS receivers on board the Astrocast constellation satellites,” Advances in Space Research, Vol 73, No1, 2024, Pages 2-19,https://doi.org/10.1016/j.asr.2023.10.001.

[2] Juraj Papco, Matus Bakon, Lukas Kubica, Gabika Belicova, Branislav Droscak, Martin Ferianc, Martin

Rovnak, Antonio M. Ruiz, Joaquim J. Sousa, Satellite-based InSAR Geodesy and Collocation with GNSS, Procedia Computer Science, Volume 239, 2024, Pages 2329-2340, https://doi.org/10.1016/j.procs.2024.06.426.

[3] Xiaoni Zheng, Yanpeng Dong, Yufei Zhao, Baosheng Zhang, Mengfan Li, “TSF-GINS: Based on time-fixed sliding window with factor graph a global navigation satellite system and inertial measurement unit tightly coupled localization system,” Measurement, Vol 239, 2025, https://doi.org/10.1016/j.measurement.2024.115421.

[4] Shengyi Xu, Qiangwen Yang, Xiaodong Du, Xiaolong Xu, Qile Zhao, Long Yang, Yanan Qin, Jing Guo, “Multi-GNSS Precise Point Positioning enhanced by the real navigation signals from CENTISPACETM LEO mission,”Advances in Space Research,Vol 73, No 8, 2024, Pages 4175-4186, https://doi.org/10.1016/j.asr.2024.01.017.

[5] Vicente Navarro, Sara del Rio, Maria del Mar Millán, Alain Messina, Javier Ventura-Traveset, GSSC Now: ESA Thematic exploitation platform for navigation digital transformation. Enhancing GNSS scientific research, Advances in Space Research, Vol 74, No 6, 2024, Pages 2728-2751, https://doi.org/10.1016/j.asr.2024.02.016.

[6] Pavithra Goravi Sukumar, Modugu Krishnaiah, Rekha Velluri, Pooja Satish, Sharmila Nagaraju, Nandini Gowda Puttaswamy, Mallikarjunaswamy Srikantaswamy, “An efficient adaptive reconfigurable routing protocol for optimized data packet distribution in network on chips”, International Journal of Electrical and Computer Engineering (IJECE), Vol 14,No.1, 2024,PP 305-314, http://doi.org/10.11591/ijece.v14i1.pp305-314.

[7] Rekha Sathyanarayana, Nataraj Kanathur Ramaswamy, Mallikarjunaswamy Srikantaswamy, Rekha Kanathur Ramaswamy, “An efficient unused integrated circuits detection algorithm for parallel scan architecture” International Journal of Electrical and Computer Engineering (IJECE), Vol 14,No.1, 2024, pp 469-478, http://doi.org/10.11591/ijece.v14i1.pp469-478

[8] Honnegowda, J., Mallikarjunaiah, K., Srikantaswamy, M. (2024). An efficient abnormal event detection system in video surveillance using deep learning-based reconfigurable autoencoder. Ingénierie des Systèmes d’Information, Vol. 29, No. 2, pp. 677-686. https://doi.org/10.18280/isi.290229.

[9] M. S, S. M D, R. S, A. K. Subramanian, V. K. V and M. S. S, "Convolutional Neural Network-based image tamper detection with Error Level Analysis," 2024 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE), Bangalore, India, 2024, pp. 1-7, doi: 10.1109/IITCEE59897.2024.10467563.

[10] T. Zhang, L. Zhou, X. Feng, J. Shi, Q. Zhang and X. Niu, "INS-Aided GNSS Pseudo-Range Error Prediction Using Machine Learning for Urban Vehicle Navigation," in IEEE Sensors Journal, vol. 24, no. 6, pp. 9135-9147, 15 March15, 2024, doi: 10.1109/JSEN.2024.3355705.

[11] Q. Zhang, H. Lin, L. Ding, Q. Chen, T. Zhang and X. Niu, "RANSAC-Based Fault Detection and Exclusion Algorithm for Single- Difference Tightly Coupled GNSS/INS Integration," in IEEE Transactions on Intelligent Vehicles, vol. 9, no. 2, pp. 3986-3997, Feb. 2024, doi: 10.1109/TIV.2023.3342274.

[12] Rekha S, Nataraj K R, Rakshitha C M, Manu K S, Rekha K R, Rohith H, “Swaraksha: Comprehensive Safety Application for Community Protection”, Vol. 46 No. 03 (2025), https://doi.org/10.52783/tjjpt.v46.i03.9882

[13] X. Meng, H. Tan, P. Yan, Q. Zheng, G. Chen and J. Jiang, "A GNSS/INS Integrated Navigation Compensation Method Based on CNN–GRU + IRAKF Hybrid Model During GNSS Outages," in IEEE Transactions on Instrumentation and Measurement, vol. 73, pp. 1-15, 2024, Art no. 2510015, doi: 10.1109/TIM.2024.3369131.

[14] Q. Chen, H. Lin, J. Kuang, Y. Luo and X. Niu, "Rapid Initial Heading Alignment for MEMS Land Vehicular GNSS/INS Navigation System," in IEEE Sensors Journal, vol. 23, no. 7, pp. 7656-7666, 1 April1, 2023, doi: 10.1109/JSEN.2023.3247587.

[15] T. Zhang, M. Yuan, L. Wang, H. Tang and X. Niu, "A Robust and Efficient IMU Array/GNSS Data Fusion Algorithm," in IEEE Sensors Journal, vol. 24, no. 16, pp. 26278-26289, 15 Aug.15, 2024, doi:

10.1109/JSEN.2024.3418383.

[16] Q. Shuai, F. Guo, G. Li, X. Zhao, B. Zhu and J. Sun, "A Dynamic Continuous Constrained Phase Factor Graph Optimization Method for GNSS Kinematic Precise Point Positioning," in IEEE Sensors Journal, vol. 23, no. 10, pp. 10739-10747, 15 May15, 2023, doi: 10.1109/JSEN.2023.3264782.

[17] Sathyanarayana, R., Ramaswamy , N. K. ., & Ramaswamy, R. K. . (2023). An Efficient Low-Power Reconfigurable Model For CMOS-Based SRAM Using FPGA. *International Journal of Intelligent Systems and Applications in Engineering*, *12*(1), 500–515. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/3948

[18] Mahendra, H.N., Mallikarjunaswamy, S. & Subramoniam, S.R. An assessment of vegetation covers of Mysuru City, Karnataka State, India, using deep convolutional neural networks. Environ Monit Assess 195, 526 (2023). https://doi.org/10.1007/s10661- 023-11140-w.

[19] Umashankar, M.L., Mallikarjunaswamy, S., Sharmila, N., Kumar, D.M., Nataraj, K.R. (2023). A Survey on IoT Protocol in Real- Time Applications and Its Architectures. In: Kumar, A., Senatore, S., Gunjan, V.K. (eds) ICDSMLA 2021. Lecture Notes in Electrical Engineering, vol 947. Springer, Singapore. https://doi.org/10.1007/978-981-19-5936-3_12.

[20] S, P., S, M. & N, S. Image region driven prior selection for image deblurring. Multimed Tools Appl 82, 24181–24202 (2023). https://doi.org/10.1007/s11042-023-14335-y.

[21] H.N. Mahendra, S. Mallikarjunaswamy and Sudalayandi Rama Subramoniam, “An assessment of built-up cover using geospatial techniques – a case study on Mysuru District, Karnataka State, India” International Journal of Environmental Technology and Management, Vol. 26, No. 3-5, 2023, pp 173-188, https://doi.org/10.1504/IJETM.2023.130787.

[22] Tengfei Bai, Hongzhou Chai, Xiangyu Tian, He Guo, Hamed Karimian, Jialong Sun, Chao Dong, “A shipboard integrated navigation algorithm based on smartphone built-in GNSS/IMU/MAG sensors,” Advances in Space Research,2024, https://doi.org/10.1016/j.asr.2024.07.048.

[23] Yingkai Cai, Yichao Li, Zhaokui Wang, Real-time high-precision baseline measurement of satellite formation flying based on GNSS, Advances in Space Research, Vol 73, No.10, 2024, Pages 5171-5187, https://doi.org/10.1016/j.asr.2024.02.025.

[24] S. Rekha and B. S. Hithaishi, "Car Surveillance and Driver Assistance Using Blackbox with the Help of GSM and GPS Technology," 2017 International Conference on Recent Advances in Electronics and Communication Technology (ICRAECT), Bangalore, India, 2017, pp. 297-301, doi: 10.1109/ICRAECT.2017.57.

[25] B. M. Kavya, N. Sharmila, K. B. Naveen, S. Mallikarjunaswamy, K. S. Manu and S. Manjunatha, "A Machine Learning based Smart Grid for Home Power Management using Cloud-Edge Computing System," 2023 International Conference on Recent Advances in Science and Engineering Technology (ICRASET), B G NAGARA, India, 2023, pp. 1-6, doi: 10.1109/ICRASET59632.2023.10419952.

[26] S. Sheela, K. B. Naveen, N. M. Basavaraju, D. Mahesh Kumar, M. Krishnaiah and S. Mallikarjunaswamy, "An efficient vehicle to vehicle communication system using intelligent transportation system," 2023 International Conference on Recent Advances in Science and Engineering Technology (ICRASET), B G NAGARA, India, 2023, pp. 1-6, doi: 10.1109/ICRASET59632.2023.10420043.