Smart E-VANET For Scheduling and Communication

Welcome Page with Sign Up and Sign In:

The Welcome Page serves as the entry point to our E-VANET platform, inviting users to explore the future of electric mobility. Choose "Sign Up" to become part of our community or "Sign In" if you already have an account.

The Creating Account module facilitates user registration for EV owners.

By completing this process, users provide essential details to access seamless services.

Email Confirmation for EV Owner:

After registering, users receive an email confirmation as part of the account verification process. This email contains a confirmation link, ensuring the security and legitimacy of their E-VANET account.

Redirecting to Login Page after Confirming Email:

Upon successfully confirming their email, users are seamlessly redirected to the Login Page. This step ensures a smooth transition, allowing users to sign in with their verified credentials.

Block chain code :

Output :

Broadcasting Message Feature for EV Owner:

The Broadcasting Message Feature empowers EV owners to share important information with the E-VANET community. Whether it's traffic updates, safety alerts, or community announcements, EV owners can utilize this feature to broadcast messages, fostering real-time communication and collaboration within the E-VANET network.

Receiving Broadcasted Messages:

EV owners seamlessly receive broadcasted messages from their peers through the E-VANET platform. Stay informed about road conditions, events, and community updates as part of our commitment to enhancing communication and safety within the electric vehicle community.

Dataset fo SOC pred module :

```
combined_df.head()
```

erative Braking Signal	Battery Voltage [V]	Battery Current [A]	...	Temperature Footwell Co-Driver [C]	Temperature Feetvent Co-Driver [C]	Temperature Feetvent Driver [C]	Temperature Head Co-Driver [C]	Temperature Head Driver [C]	Temperature Vent right [C]	Temperature Vent central right [C]	Temperature Vent central left [C]	Temperature Vent right [C]	Velocity [km/h]]]
0.0	391.4	-2.20	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
0.0	391.4	-2.21	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
0.0	391.4	-2.26	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
0.0	391.4	-2.30	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
0.0	391.4	-2.30	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1094793 entries, 0 to 1094792
Data columns (total 59 columns):
 #   Column                                Non-Null Count    Dtype
---  ------                                --------------    -----
 0   Unnamed: 0                            1094793 non-null  int64
 1   Time [s]                              1094793 non-null  float64
 2   Velocity [km/h]                       1078364 non-null  float64
 3   Elevation [m]                         1094793 non-null  float64
 4   Throttle [%]                          1094793 non-null  float64
 5   Motor Torque [Nm]                     1094793 non-null  float64
 6   Longitudinal Acceleration [m/s^2]     1094793 non-null  float64
 7   Regenerative Braking Signal           1094793 non-null  float64
 8   Battery Voltage [V]                   1094793 non-null  float64
 9   Battery Current [A]                   1094793 non-null  float64
 10  Battery Temperature [C]               10090 non-null    float64
 11  max. Battery Temperature [C]          10090 non-null    float64
 12  SoC [%]                               1064000 non-null  float64
 13  displayed SoC [%]                     1063999 non-null  float64
 14  min. SoC [%]                          1064000 non-null  float64
 15  max. SoC [%)                          1063999 non-null  float64
 16  Heating Power CAN [kW]                1094793 non-null  float64
 17  Heating Power LIN [W]                 784206 non-null   float64
 18  Requested Heating Power [W]           1094793 non-null  float64
 19  AirCon Power [kW]                     1094793 non-null  float64
...
 57  Temperature Vent right [ C]           622453 non-null   float64
 58  Velocity [km/h]]]                     16429 non-null    float64
dtypes: float64(57), int64(2)
memory usage: 492.8 MB
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

- **Used parameter :**

NN

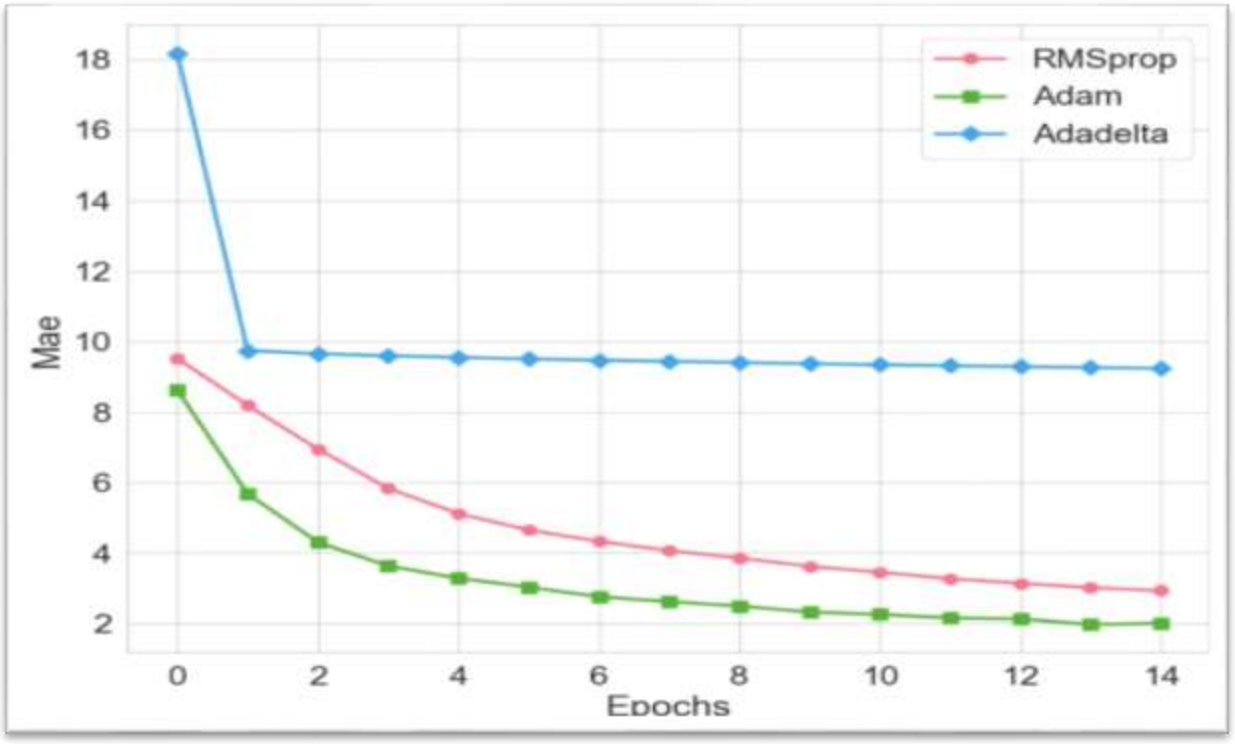
```
combined_df = combined_df.rename(columns={
    'Time [s]': 'Time',
    'Battery Voltage [V]': 'Battery Voltage',
    'Battery Temperature [ C]': 'Battery Temperature',
    'AirCon Power [kW]': 'AirCon Power',
    'Heater Signal': 'Heater Signal',
    'Ambient Temperature Sensor [ C]' : 'Ambient Temperature Sensor',
    'Temperature Head Driver [ C]' : 'Temperature Head Driver',
    'Temperature Vent right [ C]' : 'Temperature Vent right'
})
```

- **Model:**

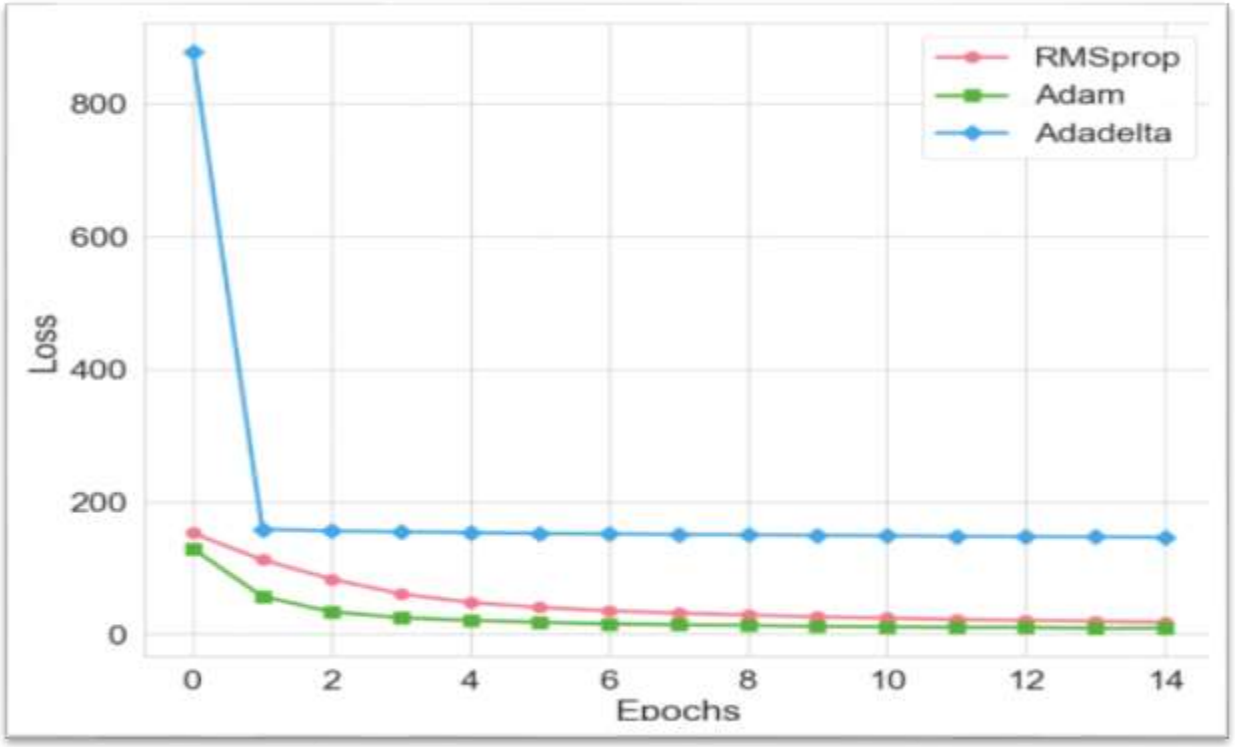
```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 6, 64)             256

 max_pooling1d (MaxPooling1  (None, 3, 64)             0
 D)

 flatten (Flatten)           (None, 192)               0

 dense (Dense)               (None, 128)               24704

 dense_1 (Dense)             (None, 1)                 129

=================================================================
Total params: 25089 (98.00 KB)
Trainable params: 25089 (98.00 KB)
Non-trainable params: 0 (0.00 Byte)
```

Training graph :



Training loss graph :



Adam prediction comparison :