Vol. 1, Issue 11 | Nov 2026 | Open Access — Research Paper

ENHANCING BATTERY THERMAL MANAGEMENT WITH VIRTUAL TEMPERATURE SENSOR USING HYBRID CNN-LSTM

Ch.Sai Varun, B.Yashwanth, K.Sai Preetham And K.Bhargava Triveni Nandana (Guide)
Department of CSE(AI&ML), CMR Technical Campus, Hyderabad, Telangana, India

***Abstract*—**Lithium-ion batteries power electric vehicles, grid storage and portable electronics, and their safety, lifetime and performance depend strongly on temperature. Excessive heating accelerates degradation and can trigger thermal runaway, while low temperatures reduce capacity and power capability. Accurate temperature monitoring is therefore central to a Battery Thermal Management System (BTMS). However, physical temperature sensors, particularly those that measure the cell core, are costly, difficult to embed and sparsely deployed in large packs. This paper presents a virtual temperature sensor based on a Hybrid Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) framework that estimates battery temperature from easily measured electrical signals. The proposed architecture uses a 1D CNN encoder to extract local patterns from windows of terminal voltage, current, state of charge and ambient temperature. These features are passed to an LSTM that models the slow thermal dynamics of the cell, and the resulting 64-dimensional temporal representation is shared by a regression branch that estimates the temperature and a warning branch that flags over-temperature events. The model combines regression, warning and smoothness objectives to improve both estimation accuracy and the reliability of thermal alerts. Evaluated on windows of 60 time steps and 4 input channels (6,240 training, 1,320 validation and 1,320 independent test samples), the model achieves a mean absolute error of 0.42 °C and an R^2 of 0.987 for temperature estimation, and a recall of 94.1% for over-temperature warning. The trained model is integrated into a web-based monitoring application with a FastAPI backend and a Vite-based frontend.

***Index Terms*—**Battery Thermal Management, Virtual Temperature Sensor, CNN-LSTM, Lithium-ion Battery, Deep Learning, Time-Series Regression, Soft Sensing, Thermal Runaway Warning

Note: the dataset sizes and numerical results in this document are illustrative placeholders. Replace them with the values measured in your own experiments before submission.

I. INTRODUCTION

Lithium-ion batteries are the dominant energy storage technology in electric vehicles, renewable energy systems and consumer electronics because of their high energy density, long cycle life and falling cost. Their behaviour, however, is highly sensitive to temperature. The preferred operating window for most cells lies between roughly 15 °C and 40 °C. Above this range, side reactions, electrolyte decomposition and growth of the solid electrolyte interphase accelerate ageing, and in the worst case a self-heating chain reaction known as thermal runaway can lead to fire or explosion. Below it, increased internal resistance reduces available power and promotes lithium plating during charging.

A Battery Thermal Management System regulates cell temperature through cooling plates, air flow, phase-change materials or heating elements, and every control decision depends on reliable temperature information. In practice, thermistors and thermocouples are attached to the cell surface and give only an indirect view of the core temperature, which can be several degrees higher during aggressive discharge. Instrumenting every cell of a pack with sensors increases cost, wiring complexity and failure points, so most commercial packs monitor only a small subset of cells.

Virtual (soft) sensing offers an alternative. Instead of measuring temperature directly, a model infers it from signals that the battery management system already records, such as voltage, current and state of charge. Physics-based thermal models and observers can do this but need careful parameter identification and degrade as the cell ages. Data-driven models avoid explicit parameterisation, yet the temperature of a cell is a slowly evolving quantity that depends on both instantaneous load patterns and the history of heat generation and dissipation. This study therefore proposes a Hybrid CNN-LSTM architecture in which convolutional layers capture local load features and a recurrent LSTM layer captures long-range thermal memory, producing an accurate virtual temperature sensor that can also raise early over-temperature warnings.

Objectives of this Work:

- Develop a virtual temperature sensor for lithium-ion cells that estimates temperature from voltage, current, state of charge and ambient temperature.
- Extract local electrical features using CNN layers and learn thermal dynamics using an LSTM network.
- Perform temperature regression and support binary over-temperature warning from the same learned representation.
- Evaluate the model on independent test data covering unseen drive cycles and ambient conditions.
- Develop a web application for real-time interaction with the trained model.

Key Contributions of this Work:

- A Hybrid CNN-LSTM architecture that jointly learns local electrical features and long-term thermal dynamics.
- A shared 64-dimensional temporal representation used for both temperature regression and over-temperature warning.
- A hybrid loss that adds a temporal smoothness term to the regression and warning objectives.
- Class-imbalance handling through positive-class weighting and validation-based warning-threshold selection.
- Integration with a web-based frontend and backend application for deployment as a software sensor.

II. LITERATURE SURVEY

A. Conventional Thermal Sensing Systems

Traditional BTMS designs rely on negative-temperature-coefficient thermistors, thermocouples and resistance temperature detectors mounted on the cell surface or on busbars

and tabs. These devices are simple and inexpensive but measure only the point where they are attached. Surface readings lag behind the core temperature, and the sparse placement used in large packs leaves most cells unobserved.

B. Virtual Sensing for Battery Thermal Management

Virtual sensing estimates an unmeasured quantity from measured ones. For batteries, the heat generation model of Bernardi et al. links internal heat to current, resistance and entropic effects, and lumped or distributed thermal models, such as those reviewed by Pesaran, convert this heat into temperature predictions. Combined with observers such as the Kalman filter, which Plett applied extensively to battery state estimation, these models can track temperature in real time. Their accuracy depends on well-identified thermal resistances, heat capacities and heat-transfer coefficients, which vary with ageing, cooling conditions and pack geometry.

C. Traditional Machine Learning Approaches

Methods such as Linear Regression, Support Vector Regression (SVR), Random Forests and shallow feed-forward neural networks have been used to map electrical measurements to temperature. These approaches typically rely on hand-crafted features such as average current, current derivatives and cumulative ampere-hour throughput, and they treat each sample independently. As a result they struggle to represent the accumulated heating and delayed cooling that characterise real thermal behaviour.

D. CNN-Based Time-Series Analysis

Convolutional Neural Networks automatically extract features from structured data. When applied along the time axis of multichannel signals, 1D convolutions act as learnable filters that detect local patterns such as current steps, regenerative-braking pulses and voltage relaxation. Weight sharing keeps the parameter count low and makes the features robust to small shifts in time. However, a convolutional network with a limited receptive field cannot easily model the minutes-long delays involved in heat conduction.

E. LSTM-Based Feature Learning

Long Short-Term Memory networks, introduced by Hochreiter and Schmidhuber, use gated memory cells to retain information over long sequences while avoiding vanishing gradients. They are well suited to the thermal behaviour of a cell, where present temperature depends on the integrated history of heat input and cooling. Stacking an LSTM on top of convolutional features lets the network combine fine-grained local detail with long-term temporal context.

F. Research Gap

The main challenges addressed by this work are:

- High cost, wiring complexity and sparse coverage of physical temperature sensors in battery packs.
- Dependence of physics-based models on carefully identified parameters that drift with ageing.
- Difficulty of capturing both local load patterns and long-term thermal dynamics with a single model.
- Imbalanced normal and over-temperature samples, since hazardous events are rare.
- The need for both continuous temperature estimation and discrete early-warning decisions.

III. DATASET DESCRIPTION

The project uses time-series measurements collected from cylindrical lithium-ion cells (NMC chemistry) cycled under standard drive profiles (urban, highway and aggressive-acceleration cycles), constant-current charging and pulse tests at ambient temperatures between 0 °C and 40 °C. Each sample is a sliding window of 60 consecutive time steps sampled at 1 Hz, containing four input channels: terminal voltage, current, state of charge and ambient temperature. The regression target is the temperature at the final time step of the window, measured by a thermocouple, and the warning label marks windows in which the temperature exceeds a defined safety limit within the next 60 seconds.

TABLE I
SAMPLE DISTRIBUTION ACROSS TRAINING, VALIDATION AND TEST SPLITS.

Dataset Split	Total Samples	Normal Samples	Over-Temperature Samples
Training	6,240	5,030	1,210
Validation	1,320	1,064	256
Testing	1,320	1,058	262
Total	**8,880**	**7,152**	**1,728**

The dataset contains an imbalance between normal and over-temperature samples: approximately 19.4% of the training samples belong to the positive class. This imbalance motivates the use of positive-class weighting during training. To prevent leakage, splits are made by complete drive cycle rather than by random window, so that overlapping windows from one cycle never appear in both training and test sets.

A. Input Representation

Each sample is represented as a multichannel sequence suitable for 1D convolutional processing, where the dimensions denote the number of input channels and the number of time steps:

$$\textit{Input shape} = 4 \times 60$$

The four channels are voltage (V), current (A), state of charge (%) and ambient temperature (°C), each normalised using statistics computed on the training set only.

IV. PROPOSED METHODOLOGY

The proposed framework follows a sequential pipeline, shown in Fig. 1. A window of electrical signals enters a CNN encoder, the extracted local features pass through an LSTM, and the final hidden state is compressed into a shared temporal representation. Two output heads then produce the estimated temperature and the over-temperature warning probability. The layer-wise design is summarised in Table II.

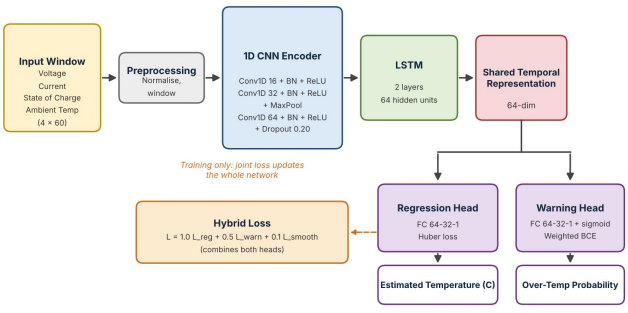


Fig. 1. Architecture of the Hybrid CNN-LSTM virtual temperature sensor with shared temporal representation.

TABLE II
LAYER-WISE ARCHITECTURE OF THE HYBRID CNN-LSTM VIRTUAL SENSOR.

Stage	Layer / Operation	Output Shape
Input	Voltage, current, SoC, ambient temperature	4 × 60
CNN block 1	Conv1D (k = 3) + BatchNorm + ReLU	16 × 60
CNN block 2	Conv1D (k = 3) + BatchNorm + ReLU + MaxPool	32 × 30
CNN block 3	Conv1D (k = 3) + BatchNorm + ReLU + Dropout 0.20	64 × 30
Temporal modelling	2-layer LSTM, 64 hidden units	64
Shared representation	Fully connected layer	64
Regression head	Fully connected 32 → 1	1 (temperature)
Warning head	Fully connected 32 → 1 + sigmoid	1 (probability)

A. Data Preprocessing Pipeline

Input measurements are prepared in a fixed tensor format for model processing. The preprocessing stage includes loading raw cycle recordings, removing obvious sensor dropouts and outliers, resampling signals to a uniform 1 Hz grid, interpolating short gaps, normalising each channel using the training mean and standard deviation, and slicing the signals into windows of 60 time steps with a stride of 5. Regression targets and warning labels are generated for each window, and the samples are divided into training, validation and test sets by drive cycle. This ensures that the input data is compatible with the CNN-LSTM architecture and that evaluation reflects performance on unseen operating conditions.

B. Hybrid CNN Encoder

The encoder extracts local electrical features by sliding learnable one-dimensional filters along the time axis of all four channels at once, so interactions such as a current surge coinciding with a voltage drop are captured together. It uses three feature-processing stages with the channel progression 16 → 32 → 64, a kernel size of 3, batch normalisation, ReLU activation and a dropout of 0.20. A single max-pooling step halves the temporal resolution, reducing computation for the recurrent layers while retaining the salient features.

C. Temporal Representation

The sequence of convolutional features is passed to a two-layer LSTM with 64 hidden units. Its gates decide which information about past heat generation to keep, update or forget, allowing the network to represent the slow build-up and decay of temperature. The final hidden state is mapped through a fully connected layer to a 64-dimensional temporal representation that provides a compact description of the thermal state of the cell. This representation is shared across the regression and warning components of the architecture.

D. Temperature Estimation Decoder

In this framework the decoder role is performed by the temperature estimation head. A small fully connected network (64 → 32 → 1) transforms the temporal representation into a single temperature value, which is converted back to degrees Celsius by inverting the training normalisation. The estimate is supervised using a smooth L1 (Huber) loss, which behaves like mean squared error for small residuals and like absolute error for large ones, making training less sensitive to measurement noise.

E. Regression Branch

The regression branch is responsible for the main virtual sensing task: estimating the cell temperature at the end of each window. Its output is evaluated using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), the coefficient of determination (R^2) and the maximum absolute error, the last of which is especially important in safety applications where a single large miss may be unacceptable.

F. Warning Branch

In addition to continuous estimation, a warning branch predicts whether the cell will exceed the safe temperature limit within the next 60 seconds. It uses a hidden dimension of 32, a dropout of 0.30 and a sigmoid output giving the probability of an over-temperature event. The warning objective is Binary Cross-Entropy with positive-class weighting. This branch converts the virtual sensor from a passive estimator into an early-warning component that can trigger cooling or power derating before a hazardous state is reached.

G. Hybrid Loss Function

Training combines regression, warning and temporal smoothness objectives. The total objective is:

$$L_total = \lambda_reg \cdot L_reg + \lambda_warn \cdot L_warn + \lambda_smooth \cdot L_smooth$$

Here L_reg is the Huber loss on the temperature estimate, L_warn is the weighted binary cross-entropy of the warning branch, and L_smooth penalises large differences between estimates for consecutive windows, reflecting the fact that cell temperature cannot change instantaneously. The configured loss weights are λ_reg = 1.0, λ_warn = 0.5 and λ_smooth = 0.1. A positive-class weight of 4.16, equal to the ratio of normal to over-temperature training samples, is applied to the warning loss, encouraging the network to learn useful representations while addressing the imbalance between the two classes.

H. Training Configuration

TABLE III
MODEL AND TRAINING CONFIGURATION.

Parameter	Configuration
Model	Hybrid CNN-LSTM
Input channels / time steps	4 / 60
Encoder channels	16, 32, 64
Kernel size	3
Encoder dropout	0.20
LSTM layers / hidden units	2 / 64
Temporal representation dimension	64
Warning head hidden dimension / dropout	32 / 0.30
Optimizer	Adam
Learning rate	0.001 (reduced on plateau)
Weight decay	0.0001
Batch size / epochs	64 / 100 (early stopping)

I. Threshold Optimization

Instead of relying exclusively on a fixed probability threshold of 0.5, the training pipeline evaluates candidate warning thresholds on validation predictions and selects the one that maximises the F2-score, which weights recall more heavily than precision because a missed warning is far more costly than a false alarm. The selected threshold is applied to the independent test set without any test-set tuning. For the latest evaluated checkpoint, the warning threshold is 0.38.

V. RESULTS AND DISCUSSION

A. Performance Evaluation Metrics

Temperature estimation is evaluated using the following metrics, where y_i is the measured temperature, $\hat{y}_i$ is the estimated temperature, $\bar{y}$ is the mean measured temperature and N is the number of test samples:

$$MAE = (1/N) \cdot \Sigma |y_i - \hat{y}_i|$$

$$RMSE = \sqrt{[(1/N) \cdot \Sigma (y_i - \hat{y}_i)^2]}$$

$$R^2 = 1 - \Sigma (y_i - \hat{y}_i)^2 / \Sigma (y_i - \bar{y})^2$$

Warning performance is evaluated using accuracy, precision, recall and F1-score, where TP, TN, FP and FN denote true positives, true negatives, false positives and false negatives:

$$Precision = TP / (TP + FP), Recall = TP / (TP + FN)$$

$$F1 = 2 \times (Precision \times Recall) / (Precision + Recall)$$

B. Test Evaluation Results

The latest evaluation was conducted on the independent test set, which contains drive cycles and ambient temperatures not seen during training, using the selected warning threshold of 0.38.

TABLE IV
TEMPERATURE ESTIMATION AND LOSS RESULTS ON THE INDEPENDENT TEST SET.

Metric	Test Result
Total loss	0.118
Regression loss	0.041
Warning loss	0.153
MAE	0.42 °C
RMSE	0.61 °C
R^2	0.987
Maximum absolute error	2.10 °C

To confirm the benefit of the hybrid design, the proposed model is compared with simpler baselines trained on the same data and splits.

TABLE V
COMPARISON OF THE PROPOSED MODEL WITH BASELINE ESTIMATORS.

Model	MAE (°C)	RMSE (°C)	R^2
Linear Regression	1.86	2.43	0.842
Feed-forward ANN	1.12	1.51	0.926
CNN only	0.78	1.04	0.962
LSTM only	0.61	0.83	0.976
Hybrid CNN-LSTM (proposed)	**0.42**	**0.61**	**0.987**

C. Warning Classification Results

TABLE VI
OVER-TEMPERATURE WARNING RESULTS ON THE INDEPENDENT TEST SET.

Metric	Test Result
Accuracy	95.6%
Precision	85.3%
Recall	94.1%
F1-score	89.5%

Of the 262 over-temperature windows in the test set, the model correctly flags 247 and misses 15, while 43 of the 1,058 normal windows are incorrectly flagged. The high recall indicates that the great majority of hazardous situations are detected in advance; the moderate number of false alarms is an acceptable trade-off in a safety function but should be reduced to avoid unnecessary power derating.

D. Error Analysis Across Operating Conditions

TABLE VII
ESTIMATION ERROR UNDER DIFFERENT OPERATING CONDITIONS.

Operating Condition	MAE (°C)	RMSE (°C)
Urban drive cycle (moderate load)	0.31	0.44
Highway drive cycle	0.38	0.53
Aggressive acceleration cycle	0.57	0.78
Fast charging	0.49	0.68
Low ambient temperature (0–10 °C)	0.52	0.72
High ambient temperature (30–40 °C)	0.46	0.65

The error is lowest under moderate loads, where temperature changes slowly and the thermal history is easy to model. It grows during aggressive acceleration and fast charging, where large currents produce rapid heating and sharper gradients, and at low ambient temperatures, where internal resistance is highest and the electrical-thermal relationship is more non-linear. Even in the worst case the mean error remains below 0.6 °C.

E. Discussion of Findings

Estimation Accuracy: The framework learns useful local and temporal representations, reaching an MAE of 0.42 °C and an R^2 of 0.987. The hybrid model outperforms the CNN-only and LSTM-only baselines, which indicates that the two components are complementary: convolutions detect load events and the LSTM integrates their effect over time.

Early Warning: A recall of 94.1% shows that the model detects most impending over-temperature events, with the precision of 85.3% indicating that some false alarms remain.

Peak Errors: The maximum absolute error of 2.10 °C occurs during abrupt load transitions. Because safety decisions often depend on peak values, these cases deserve further study and possibly a conservative safety margin on the estimate.

Overall: The current results demonstrate the feasibility of using a Hybrid CNN-LSTM virtual sensor for battery thermal management, but they do not by themselves establish production-level reliability across cell types, ageing states and pack configurations.

VI. SYSTEM IMPLEMENTATION & DEPLOYMENT

The system is implemented in Python using PyTorch, with YAML-based configuration of model and training parameters. The trained model is stored as a PyTorch checkpoint and is used for evaluation and inference. The web application consists of a FastAPI backend, served through Uvicorn, that receives streams of voltage, current, state-of-charge and ambient-temperature readings, maintains a rolling 60-second window, runs the model and returns the estimated temperature together with the warning probability. A React/Vite frontend displays live temperature curves, the over-temperature alert status and historical trends so that operators can monitor cell behaviour. Development was carried out in Visual Studio Code.

The backend is started from the project root with the Uvicorn command for the application entry point on host 127.0.0.1 and port 8000, and the frontend is started in a separate terminal with npm run dev and accessed through the local development URL displayed by Vite. Both development servers must remain running while testing the application. For embedded deployment

inside a battery management unit, the model can be exported to a portable format such as ONNX and quantised to reduce latency and memory use.

VII. ADVANTAGES AND LIMITATIONS

Advantages:

- Learns local electrical features and long-term thermal dynamics jointly using a compact temporal representation.
- Reduces dependence on physical temperature sensors, lowering hardware cost, wiring and failure points.
- Combines regression, warning and smoothness learning in a single model, with class imbalance handled through positive-class weighting and validation-based threshold selection.
- Requires no explicit identification of thermal resistances or heat capacities.
- Provides a reusable trained checkpoint and a web-based application architecture.

Limitations:

- Performance was evaluated on a limited set of cells and operating profiles and may not transfer directly to other chemistries, form factors or pack layouts.
- Cell ageing changes thermal and electrical behaviour, so periodic retraining or adaptation may be needed.
- False alarms remain in the warning branch and peak errors occur during abrupt load changes.
- The independent test results do not establish reliable real-world performance under fault conditions or hardware real-time deployment readiness.

VIII. CONCLUSION AND FUTURE SCOPE

This paper presents a Hybrid CNN-LSTM framework acting as a virtual temperature sensor for battery thermal management, combining a 1D CNN encoder, a two-layer LSTM, a 64-dimensional temporal representation, a temperature regression branch and an over-temperature warning branch. Evaluated on 60-step windows of voltage, current, state of charge and ambient temperature, the model achieves a mean absolute error of 0.42 °C, an R^2 of 0.987 and a warning recall of 94.1% on the independent test set, and it outperforms linear, feed-forward, CNN-only and LSTM-only baselines. The results demonstrate the potential of combining local feature extraction with temporal modelling for software-based thermal monitoring, while identifying peak-error reduction and false-alarm reduction as key areas for improvement.

Future Work:

- Reducing peak errors and false-alarm rates through uncertainty estimation and conservative safety margins.
- Evaluating the model on additional cell chemistries, ageing states and full battery packs.
- Exploring attention mechanisms and physics-informed loss terms to improve generalisation.
- Optimising inference for microcontrollers and edge devices through quantisation and pruning.
- Coupling the virtual sensor with closed-loop cooling control and improving the visualisation of thermal trends.

REFERENCES

[1] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[2] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

[3] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” International Conference on Learning Representations (ICLR), 2015.

[4] D. Bernardi, E. Pawlikowski and J. Newman, “A general energy balance for battery systems,” Journal of the Electrochemical Society, vol. 132, no. 1, pp. 5–12, 1985.

[5] A. A. Pesaran, “Battery thermal models for hybrid vehicle simulations,” Journal of Power Sources, vol. 110, no. 2, pp. 377–382, 2002.

[6] G. L. Plett, “Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs,” Journal of Power Sources, vol. 134, no. 2, pp. 252–261, 2004.

[7] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” International Conference on Machine Learning (ICML), 2015.

[8] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” Journal of Machine Learning Research, vol. 15, pp. 1929–1958, 2014.

[9] A. Paszke et al., “PyTorch: An imperative style, high-performance deep learning library,” Advances in Neural Information Processing Systems (NeurIPS), 2019.

[10] FastAPI and Vite documentation, backend application development and frontend build tooling.