FIRE PREVENTION STRATEGIES IN ELECTRIC VEHICLE BATTERIES USING MACHINE LEARNING

by

Umit SENYUREK[a*], Muhammed Fatih PEKSEN[b], Cenk CELIK[c], and Hakan Serhad SOYHAN[d]

[a] Institute of Science and Technology, Kocaeli University, Kocaeli, Turkey
[b] Emergency and Disaster Management, Adapazarı Vocational School, Sakarya University, Sakarya, Turkey
[c] Department of Mechanical Engineering, Kocaeli University, Kocaeli, Turkey
[d] Department of Fire Safety and Mechanical Engineering, Sakarya University, Sakarya, Turkey

Original scientific paper
https://doi.org/10.2298/TSCI2504955S

The increasing use of electric vehicles necessitates robust safety measures, particularly in battery management systems. This study emphasizes predictive maintenance by introducing a proactive approach to fire safety management using machine learning. The behavior of 60% Nickel, 20% Manganese, and 20% Cobalt (NMC 622) prismatic cells under mechanical impact was investigated, with CO and CO_2 gas emissions monitored as early indicators of thermal runaway, a phenomenon that can lead to rapid and uncontrollable temperature increases if undetected. A real-scale experimental set-up simulated mechanical impacts, and the collected data were analyzed using MATLAB to derive meaningful insights. Four machine learning models – coarse tree, binary GLM-LR, efficient linear support vector machines, and Gaussian Naive Bayes – were trained and validated to predict the likelihood of thermal runaway based on gas emission patterns. This proactive approach enhances battery reliability and safety by enabling early intervention in critical areas, ensuring passenger safety. By addressing a significant gap in current research, this study contributes to the development of smarter and safer electric vehicles.

Key words: *thermal runaway, predictive maintenance, electric vehicle batteries, machine learning models, battery management system*

Introduction

The rapidly increasing use of electric vehicles has made the safety and reliability of battery systems critical. In particular, thermal runaway is one of the biggest risks, as it can lead to fires or explosions due to uncontrolled temperature increase in battery cells [1]. In lithium-ion batteries (LIB), this is associated with mechanisms such as deformation of the SEI layer [2], reactions between the electrolyte and the electrode [3], and reactions between the binder material and the electrode [4], which can cause a short circuit [5]. During thermal runaway, the temperature can exceed 1000 °C and toxic gases can be released [6]. This poses a serious fire risk due to the high energy density batteries in electric vehicles carrying a higher fire load compared to conventional vehicles [7]. The release of flammable gases such as hydrogen, methane, and CO during thermal runaway makes fires more difficult to control [8].

* Corresponding author, e-mail: umitsenyurek@yahoo.com

This study aimed to predict the probability of thermal runaway by monitoring the CO and CO_2 gas emissions when NMC 622 prismatic cells were subjected to mechanical impacts. The gas emissions were recorded in a real-scale experimental set-up and analyzed with MATLAB. The analysis results were applied to four machine learning (ML) models (Coarse tree, Binary GLM-LR, Efficient linear support vector machines – SVM, Gaussian Naive Bayes – GNB to predict the probability of thermal runaway. The performance of the models was compared and the most effective method for early detection and intervention was determined. The study aims to provide early warning and preventive safety in critical areas by integrating these prediction models into the management system of electric vehicles batteries. This proactive approach increases the reliability and safety of batteries, ensuring longer life and safety of electric vehicles. This approach aims to fill an important gap in the existing literature in the field of battery safety [9].

Literature review

Artificial intelligence (AI), emerging in the 1950's, has yet to achieve a universal definition in literature. Broadly, AI is a branch of computer science that examines the ability of machines to perform tasks considered intelligent when done by humans. John McCarthy defines AI as machines capable of behaving like humans, while Nabiyev views it as the ability to perform cognitive processes such as reasoning, learning, and generalization [10]. The foundation of AI was laid by Turing's 1950 question, *Can machines think?* and the term was formalized at the Dartmouth Conference in 1956 [11, 12]. Early studies focused on symbolic reasoning, while expert systems became prominent in the 1980's, followed by ML and neural networks in the 1990's [13, 14]. Today, AI continues to advance rapidly, with significant progress in deep learning, reinforcement learning, and natural language processing [15].

Machine learning

The ML, a subset of AI, focuses on enabling computer systems to perform tasks by learning from data without explicit programming. Introduced in the 1980's, ML relies on training data to create statistical models, with applications like spam detection and image recognition [10]. The ML comprises supervised, unsupervised, and semi-super-vised learning. In this study, supervised ML models used and explained at the below.

Supervised learning: This method uses labeled data for training, enabling models to predict outputs for new inputs. It is widely applied in tasks such as anomaly detection in cybersecurity and disease diagnosis in healthcare [16, 17]. Supervised learning involves training and testing phases to optimize model accuracy using labeled datasets [18]. The future of AI and ML involves developing more advanced algorithms to tackle complex problems efficiently while addressing limitations like computational demands and data quality [19, 20].

Coarse tree

Decision trees were developed by William Belson and Ross Quinlan in the 1960's. They are used in fields such as data mining, finance, healthcare, and marketing for classification and regression problems. The advantages of decision trees include their ease of understanding and interpretation, computational efficiency, and ability to handle missing data [21].

The Coarse tree model is a type of decision tree that simplifies the decision-making process by using fewer splits, making it fast and easy to train. This model is particularly preferred in scenarios where interpretability and speed are crucial, and it enhances computational efficiency when working with large datasets. It classifies data based on simple decision rules derived from threshold values of specific features and has been successfully applied in various fields, such as classifying high-strength concrete mix designs and diagnosing and managing plant diseases in agriculture. However, its simplicity can limit performance on complex datasets, and its accuracy depends on the size and quality of the dataset. To improve performance, techniques like cross-validation, integration with more complex methods, and parallel processing can be employed [22, 23].

Binary generalized linear model logistic regression

Logistic regression was introduced by David Cox in 1958. It is used in various fields such as medicine, social sciences, economics, and engineering to predict binary outcomes. The advantages of logistic regression include its effectiveness in predicting binary outcomes, high interpretability of the model, and reduction of overfitting risk [24].

Binary GLM-LR is a statistical method used for binary classification problems where the outcome variable has two possible outcomes. This model estimates the probability of a given input belonging to a particular class. It is widely used in fields such as medical research, social sciences, and engineering. In medical research, it is commonly used to predict the presence or absence of a disease based on various predictors. The model estimates the probability of the dependent variable through the logit transformation of the independent variables. This method retrieves true conditional probabilities when the independent variables are in a log-linear form [25]. The model has been successfully applied in various fields. For instance, it has been used in medical diagnostics to predict the presence or absence of diseases. In one study, it achieved high accuracy rates in predicting diseases such as chronic kidney disease [26]. Additionally, it has been used in social sciences to predict the likelihood of individuals exhibiting certain behaviors. However, the model has some limitations. Specifically, the assumption of conditional independence of independent variables may not always hold true in real-world data, which can negatively impact the model performance. Moreover, the accuracy of the model depends on the size and quality of the dataset [27].

Efficient linear support vector machines

The SVM was developed by Vladimir Vapnik and his team in 1992. They are used in various fields such as text classification, image recognition, bioinformatics, and many others. The advantages of SVM include its effectiveness in high-dimensional data, high overall performance, and reduction of overfitting risk [28]. Efficient linear SVM is an ML algorithm that is a variant of the standard SVM, providing high accuracy and efficiency in classification problems while maintaining classification accuracy. This model is particularly effective in high-dimensional spaces and has been applied in software reliability prediction, outperforming other algorithms. Efficient linear SVM focuses on optimizing computational efficiency, especially when working with large and high-dimensional data sets. It uses various optimization techniques to reduce computational costs. This model finds linear separating hyperplanes using linear algebra and optimization methods. Specifically, it addresses multi-objective optimization problems aiming to balance multiple objectives such as classification error and the number of non-zero elements in the separating hyperplane [29].

Efficient linear SVM has been successfully applied in various fields, such as large data scenarios where subsampling methods reduce training samples for faster problem-solving without significant accuracy loss, and in agriculture for energy-efficient UAV management [30, 31]. However, it faces challenges with computational complexities in large-scale training sets and high-dimensional datasets where subsampling methods become less effective. Improvements like least squares-based SVM and multi-objective optimization approaches enhance computational speed and balance feature selection with model complexity [29, 30]. Despite limitations, efficient linear SVM remains a key algorithm, with ongoing research improving its performance and applicability for complex classification problems.

Gaussian Naive Bayes

Naive Bayes classifiers were introduced in the 1960's, and GNB is a variation of these classifiers. They are used in various fields such as text mining, spam filtering, medical diagnosis, and many others. The advantages of Naive Bayes classifiers include their simplicity and speed, good performance even with small datasets, and effectiveness under the assumption of feature independence [32]. The GNB is a probabilistic classifier based on Bayes' theorem, if features follow Gaussian distribution. This model is simple yet powerful, particularly effective when the normal assumption is held. The GNB models have been used to analyze patient data to predict the presence of diseases such as chronic kidney disease [26]. In text classification, GNB is used to categorize documents based on their content by assuming that word frequencies follow a Gaussian distribution [31]. Additionally, GNB assists in image recognition tasks by classifying images based on pixel intensity values [33]. The assumption that data follows a Gaussian distribution simplifies the calculation of probabilities, making GNB a computationally efficient and straightforward model. The probability of belonging to a particular class is calculated using the Gaussian probability density function [26].

Despite its advantages, GNB has several limitations. One major challenge is the assumption of feature independence, which may not hold true in real-world data, leading to suboptimal performance when features are correlated. Additionally, the assumption of Gaussian distribution for continuous features may not always be valid, necessitating techniques such as data transformation or discretization [31]. To address these challenges, various improvements and variants of GNB have been proposed. For example, the Three-way incremental Naive Bayes classifier combines incremental learning with three-way decision theory to handle dynamic data and improve classification performance [26]. Another approach involves using fuzzy discretization to convert continuous features into categorical ones, enhancing the model robustness [31].

Performance evaluation

In ML, four key metrics are used to evaluate the performance of classification models: accuracy, precision, recall, and F1 score. These metrics are important for analyzing the model predictive capabilities and classification success. Table 1 used for these values derived from the confusion matrix is given.

Table 1. Confusion matrix for two-class classification

Actual	Positive	Negative
Predicted positive	*TP*	*FP*
Predicted negative	*FN*	*TN*

Table 1 shows the true positives, *TP*, false positives, *FP*, false negatives, *FN*, and true negatives, *TN*, values of the confusion matrix used in binary classification. In this study, normal, *N*, conditions represent standard situations, while Stage 1, S1, represents smoke-forming and pre-ignition, conditions. If the model prediction is positive and the actual condition is abnormal, it is a *TP*. If the prediction is positive but the actual condition is normal, it is an *FP*. If the prediction is negative but the actual condition is abnormal, it is an *FN*. If the prediction is negative and the actual condition is normal, it is a *TN*.

Accuracy is the most common evaluation metric for classification models and measures the ratio of correct predictions (*TP* and *TN*) to total predictions. It is calculated:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Precision represents the proportion of positive predictions that are actually correct (*TP*) and reflects the accuracy of the model positive predictions. It is calculated:

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

Recall measures the percentage of actual anomalies correctly identified by the model, emphasizing its importance in scenarios where missing positives are costly, such as detecting abnormal conditions in electrical panels. It is calculated:

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

The *F*1 score, the harmonic mean of precision and recall, is useful for imbalanced datasets, ranging from 0 to 1, with higher values indicating better performance. It balances precision and recall, minimizing *FN* while ensuring correct positive predictions, and is calculated;

$$F1score = 2\frac{Precision \cdot Recall}{Precision + Recall} \tag{4}$$

The classification model performance, evaluated using accuracy, precision, recall, and F1 score, is detailed above. The following section presents and discusses experimental results, focusing on normal conditions as well as smoke-forming and pre-ignition conditions.

Thermal runaway phases in LIB and heat release dynamics

Understanding the thermal dynamics of LIB during fire incidents is critical for evaluating their thermal runaway behavior. Figure 1 provides an illustrative graph of the heat release rate, *HRR*, *vs.* time during a fire [34].

The *x*-axis represents time, showing the progression of the fire from its initiation until it is completely extinguished at the point labeled total time. The *y*-axis represents the heat release coefficient, indicating the amount of energy released during different fire stages. The fire stages presented in fig. 1 further divides the thermal behavior into additional regions (*N*, S1, S2, S3, and S4) to provide a comprehensive understanding [34]. The graphs underscore the importance of early detection mechanisms for LIB, particularly in the *N* and S1 regions, where thermal runaway can still be mitigated. These

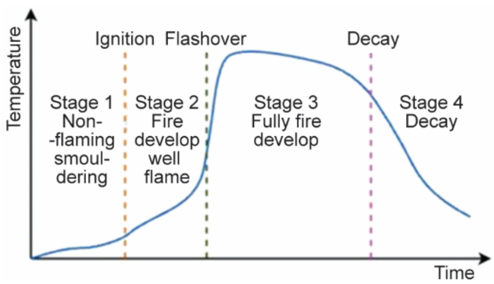

Figure 1. Different phases in the development of a fire

phases are critical for implementing predictive maintenance strategies and preventing catastrophic battery failure.

Experiment design

For this study, a specially designed 1 m^3 metal cabin with an openable lid, as shown in fig. 2(a), was used to conduct combustion experiments. These experiments were carried out using 3.5 V batteries, as depicted in fig. 2(b). These batteries are identical to those used in vehicles and are arranged sequentially. The experiments aimed to collect data to evaluate the operating conditions and safety of the batteries.

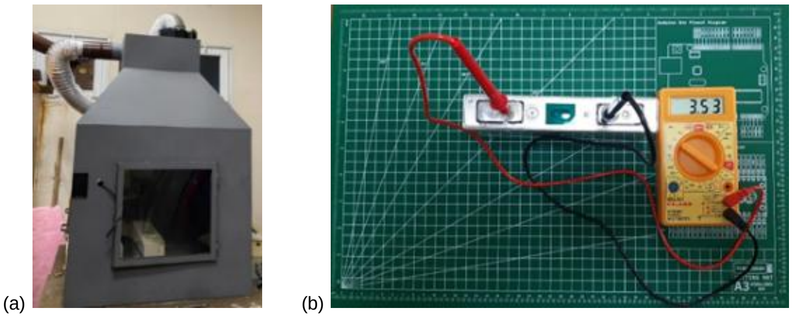

Figure 2. (a) Experiment cabinet and (b) 3.5 V battery

Figure 2(a) shows the cabinet that specially designed to analyze the gases, temperature changes and safety risks generated during the experiment. Figure 2(b) shows the battery, has the same characteristics as the batteries used in vehicles and was used sequentially in experiments. The image is given to detail the physical appearance of the batteries used in the experiments. In this study, a plug-in hybrid electric vehicle battery with a prismatic structure, NMC 622 type, 25 Ah capacity and 3.7 V nominal voltage was used.

In the process, the gases that batteries can produce in their operating environment, temperature fluctuations, and general safety risks were identified, and analyses were conducted based on the measured data. The data obtained provided a critical foundation for assessing battery operating conditions, detecting potential hazards caused by failure or degradation in advance, and preventing undesirable events such as fires or explosions. During the experiment, various sensors and devices were used to collect comprehensive data on potential hazards and environmental conditions that may arise from the battery. The data obtained with the Arduino Mega microcontroller is: The MQ4 sensor measured the amount of methane gas formed during the experiment and its rate of change and monitored the battery's potential to emit methane gas. The MQ7 sensor detected CO gas, allowing dangerous situations such as fire or battery failure to be detected. The MQ9 sensor detected flammable gases and monitored the amount and rate of change of explosive or flammable gases in the environment. The MQ135 sensor, which was used to assess air quality, assessed the safety of the environment by measuring CO_2 density. In addition, the DHT22 sensor provided data to understand the effects of these conditions on battery safety by measuring the temperature and humidity of the environment. The data collected from the sensors listed above was recorded every 2 seconds and stored with a time stamp. In this way, the changes in the data over time were analyzed in detail. Additionally, temperature and humidity data outside the cabin were collected using another DHT22 sensor using the Raspberry Pi 4 minicomputer. At the same time, surface temperature data and thermal images of the battery used in the experiment were also recorded.

The images were obtained with the MLX90640 IR thermal imaging camera. This camera records the temperature distribution both as numerical data in degrees Celsius and as thermal images, allowing detailed observation of temperature changes. The way the data was collected during the experiments is given in fig. 3.

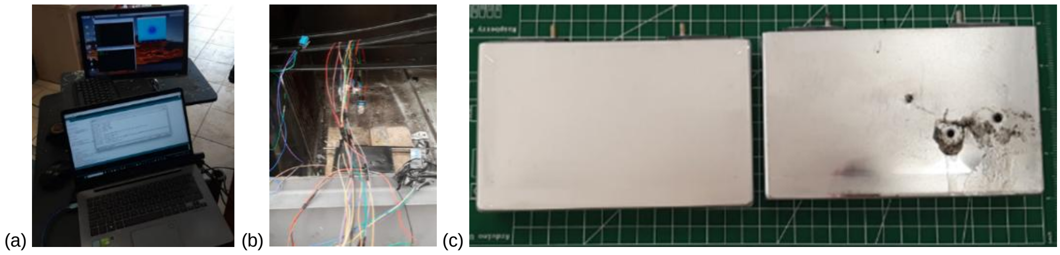

Figure 3. (a), (b) computers and cabinet inside with sensors and thermal camera, and (c) normal condition battery and drilled battery

Figure 3(a), shows the Raspberry Pi 4 minicomputer and screen (on top) and laptop computer equipment used during the experiment. Raspberry Pi 4 was used to process and display thermal images obtained by the MLX90640 IR thermal imaging camera. The thermal image data is clearly visible on the screen. The laptop on the bottom was used to process and record sensor data collected with the Arduino Mega microcontroller. This computer provides live monitoring and saves data of CH_4, CO, temperature, humidity and time stamp provided by the sensors. Figure 4(b), shows the sensors such as MQ4, MQ7, MQ9, MQ135, and DHT22 are conveniently placed within the cabin to measure gases produced by the battery, temperature changes and ambient conditions. Figure 3(c), shows the normal condition battery and drilled battery that the experiment was carried out by drilling the battery. The battery on the right displays visible signs of damage, including burn marks and structural deformation, indicating the impact of high temperatures and gases generated during the experiment. This comparison highlights the effects of extreme conditions on battery integrity and is critical for understanding safety risks associated with battery failure or malfunction in real-world scenarios. Table 2 shows the thermal images at different experiments.

Table 2. The LIB temperature display with thermal imaging at different time zones

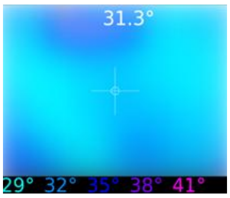	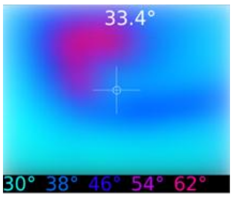	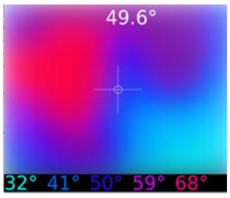	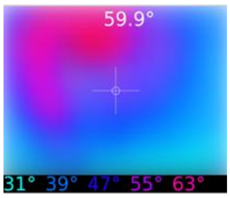
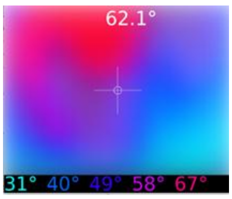	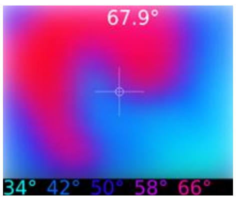	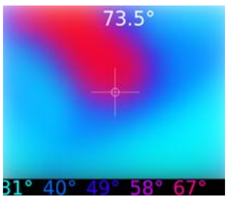	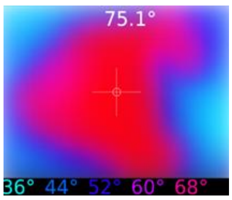

Table 2 presents eight different images showing the temperature distribution of the battery surface recorded by the thermal camera during the experiment. Each frame represents the temperature values measured at different time intervals during various experiments. The progression of the battery surface heating over time is observed. In total, in the experiments, 1910 different data sets were collected. Each data line contains 11 different features.

Experimental results

In this study, a dataset consisting of 1909 samples was used. For each algorithm, models were created using 5-fold cross-validation. In tab. 3, the algorithms were tested on the dataset using the best-selected hyperparameters. In this process, 90% of the dataset (1719 samples) was allocated for training and 10% (190 samples) for testing. Operations on the dataset and the application of ML models were performed using MATLAB R2023a academic software. The confusion matrices obtained for each algorithm are presented in fig. 4.

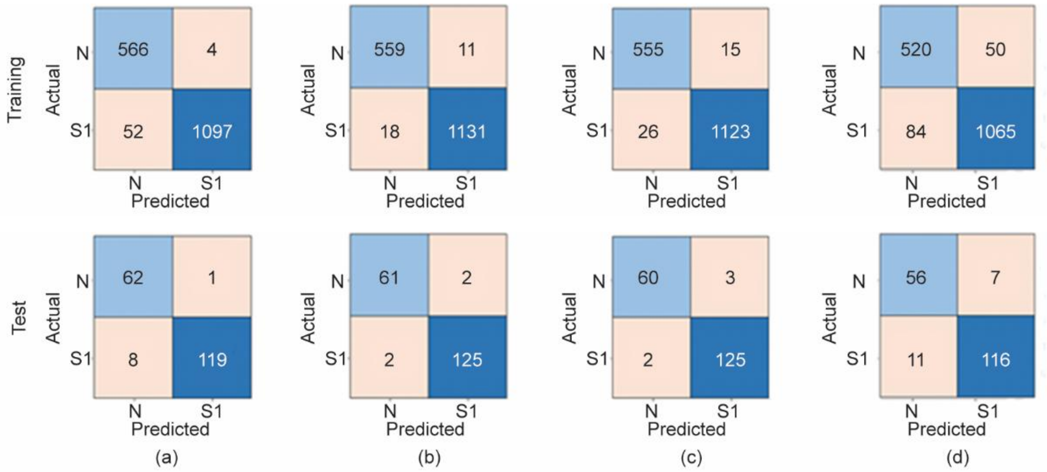

Figure 4. Confusion matrices for training and test results of used algorithms; (a) coarse tree, (b) binary GLM-LR, (c) efficient linear SVM, and (d) GNB

Figure 4 represents the performance of Coarse tree, Binary GLM-LR, Efficient linear SVM, and GNB algorithms in the training and test stages with confusion matrices. In the analysis made with the training data, Coarse tree and Binary GLM-LR models achieved high accuracy rates by exhibiting the best performance in positive (S1) and negative (*N*) classes. While the efficient linear SVM model showed balanced success, GNB made more errors compared to the other models. In the test data, a slight decrease was observed in the performance of all models compared to the training stage. However, Coarse tree and Binary GLM-LR models provided the highest accuracy in the test stage, while efficient linear SVM again offered a balanced performance. The GNB, on the other hand, showed lower success compared to the other models in both the training and test stages. In general, Coarse tree and Binary GLM-LR models stood out as the most successful algorithms. Table 3 presents the performance metrics of the ML models used in ASD on the training data (training data) based on class (*N* and S1).

Table 3. Performance metrics for training data

Model	Class	F1 Score	Precision	Recall	Accuracy [%]
Coarse tree	*N*	0.9529	0.9159	0.9930	96.74
Coarse tree	S1	0.9751	0.9964	0.9547	96.74
Binary GLM-LR	*N*	0.9747	0.9807	0.9688	98.31
Binary GLM-LR	S1	0.9873	0.9904	0.9843	98.31
Efficient linear SVM	*N*	0.9644	0.9553	0.9737	97.61
Efficient linear SVM	S1	0.9821	0.9868	0.9774	97.61
GNB	*N*	0.8859	0.8609	0.9123	92.20
GNB	S1	0.9408	0.9552	0.9269	92.20

As shown in tab. 3, Binary GLM-LR and efficient linear SVM demonstrated the best performance in both *N* and S1 classes. Coarse tree performed particularly well in the S1 class but showed slightly lower precision in the *N* class. On the other hand, GNB exhibited lower performance in both classes compared to the other models. Table 4 shows performance metrics for test results.

Table 4. Performance metrics for test data

Model	Class	F1 Score	Precision	Recall	Accuracy (%)
Coarse tree	*N*	0.8859	0.8609	0.9123	92.20
Coarse tree	S1	0.9408	0.9552	0.9269	92.20
Binary GLM-LR	*N*	0.9683	0.9683	0.9683	97.89
Binary GLM-LR	S1	0.9843	0.9843	0.9843	97.89
Efficient linear SVM	*N*	0.96	0.9677	0.9524	97.37
Efficient linear SVM	S1	0.9804	0.9766	0.9843	97.37
GNB	*N*	0.8615	0.8358	0.8889	90.53
GNB	S1	0.9280	0.9431	0.9134	90.53

As shown in tab. 4, Binary GLM-LR, and efficient linear SVM showed the best performance in both *N* and S1 classes on the test data, with high F1 scores, precision, and recall values ($\geq$ 96%). Coarse tree performed well in the S1 class but exhibited slightly lower metrics in the *N* class, particularly in precision. The GNB demonstrated the lowest performance among the models, especially in the *N* class.

Finally, fig. 5 graphically shows the accuracy on both training and testing data.
Figure 5(a) shows training data accuracy values and fig. 5(b) shows test data accuracy values used ML models in the study. Binary GLM-LR has the highest accuracy rate on both data sets and stands out as the most successful model. Efficient linear SVM showed a performance close to Binary GLM-LR, providing 97.61% accuracy on the training data and 97.37% accuracy on the test data, and became the second-best model. Coarse tree showed a moderate performance compared to the Binary GLM-LR and Efficient linear SVM models. The GNB had

the lowest accuracy rate and was evaluated as the weakest model among all. As a result, the best performance was seen in Binary GLM-LR and Efficient linear SVM models, while GNB showed the lowest performance.

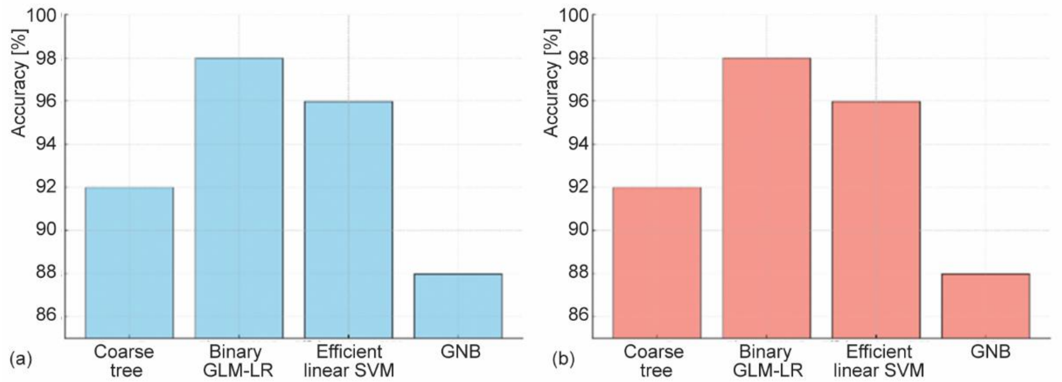

Figure 5. Accuracy comparison for; (a) training and (b) test ML models

Conclusions

This study presents an ML-based approach to prevent fire risks in LNB of electric vehicles. The study examines the gas emission behaviors of NMC 622 prismatic batteries subjected to mechanical impacts and demonstrates that CO and CO_2 gases can be used as thermal runaway indicators. Real-scale experiments have provided critical data to detect potential hazards in advance. The obtained data were applied to Coarse tree, Binary GLM-LR, Efficient linear SVM, and GNB learning models. As a result of the training, Binary GLM-LR and Efficient linear SVM models exhibited the highest accuracy rates and the best performance on both training and test data. In particular, Binary GLM-LR stood out as the most successful model by consistently providing the best results. In contrast, the GNB model showed relatively low performance. These results show that ML models can be effectively used to develop early warning mechanisms and proactively prevent thermal runaway risks.

In conclusion, the findings show that the integration of ML models into electric vehicle battery management systems can make a significant contribution to the prevention of potential thermal runaway events through early warning mechanisms. This approach can help prevent serious accidents such as fire and explosion by increasing battery reliability and passenger safety. The study provides significant progress in the field of electric vehicle battery safety, both theoretically and practically. In addition, the integration of data collection techniques such as gas emission models and thermal imaging allows continuous monitoring of battery safety status. Comparison of the performances of different ML models allowed the determination of the most effective algorithms, and these findings paved the way for innovative technologies that will further improve battery safety by providing a basis for future research. Such innovative approaches contribute to the development of sustainable and safe electric vehicle technologies.

References

[1] Lu, L., *et al*., A Review on the Key Issues for Lithium-Ion Battery Management in Electric Vehicles, *Journal of Power Sources*, *226* (2013), Mar., pp. 272-288

[2] Wang, H., *et al*., Experimental Study on the Cell-Jet Temperatures of Abused Prismatic Ni-Rich Automotive Batteries under Medium and High States of Charge, *Applied Thermal Engineering, 202* (2022), 117859
[3] Chen, J., *et al*., Designing an Intrinsically Safe Organic Electrolyte for Rechargeable Batteries, *Energy Storage Materials, 31* (2020), Oct., pp. 382-400
[4] Park, S.-Y., *et al*., Observation and Modeling of the Thermal Runaway of High-Capacity Pouch Cells due to an Internal Short Induced by an Indenter, *Journal of Energy Storage, 72* (2023), 108518
[5] Wang, E., *et al*., Safety Assessment of Polyolefin and Nonwoven Separators Used in Lithium-Ion Batteries, *Journal of Power Sources, 461* (2020), 228148
[6] Golubkov, A. W., *et al*., Thermal Runaway of Commercial 18650 Li-Ion Batteries with LFP and NCA Cathodes - Impact of State of Charge and Overcharge, *RSC Adv., 5* (2015), 70, pp. 57171-57186
[7] Sun, P., *et al*., A Review of Battery Fires in Electric Vehicles, *Fire Technology, 56* (2020), Jan., pp. 1361-1410
[8] Jin, Y., *et al*., Explosion Hazards Study of Grid-Scale Lithium-Ion Battery Energy Storage Station, *Journal of Energy Storage, 42* (2021), 102987
[9] Ghiji, M., *et al.*, A Review of Lithium-Ion Battery Fire Suppression. *Energies, 13* (2020), 19, 5117
[10] Bingol, K., *et al*., Artificial Intelligence Applications in Earthquake Resistant Architectural Design: Irregular Load-Bearing System Detection Using Deep Learning and Image Processing Method, *Journal of the Faculty of Engineering and Architecture of Gazi University, 35* (2020), 4, pp. 2197-2209
[11] Taye, M. M., Understanding of Machine Learning with Deep Learning: Architectures, Workflow, Applications and Future Directions, *Computers, 12* (2023), 5, 91
[12] Hirani, R., *et al.*, Artificial Intelligence and Healthcare: A Journey through History, Present Innovations, and Future Possibilities, *Life, 14* (2024), 5, 557
[13] Gerlich, M., Perceptions and Acceptance of Artificial Intelligence: A Multi-Dimensional Study, *Social Sciences, 12* (2023), 9, 502
[14] Khan, F. H., *et al*., Advancements in Microprocessor Architecture for Ubiquitous AI – An Overview on History, Evolution, and Upcoming Challenges in AI Implementation, *Micromachines, 12* (2021), 6, 665
[15] Valente, J., *et al*., Developments in Image Processing Using Deep Learning and Reinforcement Learning, *J. Imaging, 9* (2023), 10, 207
[16] Rafique, S. H., *et al*., Machine Learning and Deep Learning Techniques for Internet of Things Network Anomaly Detection – Current Research Trends, *Sensors, 24* (2024), 6, 1968
[17] Roy, S., *et al*., Demystifying Supervised Learning in Healthcare 4.0: A New Reality of Transforming Diagnostic Medicine, *Diagnostics, 12* (2022), 10, 2549
[18] Wang, J., *et al*., Multi-Augmentation-Based Contrastive Learning for Semi-Supervised Learning, *Algorithms, 17* (2024), 3, 91
[19] Van Engelen, J. E., Hoos, H. H., A Survey on Semi-Supervised Learning, *Machine Learning, 109* (2020), 2, pp. 373-440
[20] Zhou, X., *et al*., Study on the Evolution of Spatiotemporal Dynamics and Regional Differences in the Development of Digital Agriculture in China, *Sustainability, 16* (2024), 16, 6947
[21] Cao, Y., Huang, X., A Coarse-to-Fine Weakly Supervised Learning Method for Green Plastic Cover Segmentation Using High-Resolution Remote Sensing Images, *ISPRS Journal of Photogrammetry and Remote Sensing, 188* (2022), June, pp. 157-176
[22] Alghamdi, S. J., Classifying High Strength Concrete Mix Design Methods Using Decision Trees, *Materials, 15* (2022), 5, 1950
[23] Men, X., *et al.*, Site Class Effects on Stump and Coarse Root Biomass Models of Larix Olgensis in Northeastern China, *Forests, 13* (2022), 8, 1259
[24] Gomila, R., Logistic or Linear? Estimating Causal Effects of Experimental Treatments on Binary Outcomes Using Regression Analysis, *Journal of Experimental Psychology: General, 150* (2020), 4, 700
[25] Schaeben, H., (2014). Targeting: Logistic Regression, Special Cases and Extensions, *ISPRS International Journal of Geo-Information, 3* (2014), 4, pp. 1387-1411
[26] Yang, Z., *et al*., A New Three-Way Incremental Naive Bayes Classifier, *Electronics, 12* (2023), 7, 1730
[27] Barros, W. K. P., *et al*., Fully Parallel Proposal of Naive Bayes on FPGA, *Electronics, 11* (2022), 16, 2565
[28] Lei, Z., Lan, L., Improved Subsampled Randomized Hadamard Transform for Linear SVM, *Proceedings of the AAAI Conference on Artificial Intelligence, 34* (2020), 4, pp. 4519-4526

[29] Pirouz, B., Pirouz, B., Multi-Objective Models for Sparse Optimization in Linear Support Vector Machine Classification, *Mathematics, 11* (2023), 17, 3721
[30] Bárcenas, R., *et al*., On Subsampling Procedures for Support Vector Machines, *Mathematics, 10* (2022), 20, 3776
[31] Al-Naeem, M., *et al*., Support Vector Machine-Based Energy Efficient Management of UAV Locations for Aerial Monitoring of Crops over Large Agriculture Lands, *Sustainability, 15* (2023), 6421
[32] Ali, L., *et al*., A Feature-Driven Decision Support System for Heart Failure Prediction Based on $\chi 2$ Statistical Model and Gaussian Naive Bayes, *Computational and Mathematical Methods in Medicine, 2019* (2019), 6314328
[33] Zhang, Y., *et al.*, Intelligent Identification for Rock-Mineral Microscopic Images Using Ensemble Machine Learning Algorithms, *Sensors, 19* (2019), 18, 3914
[34] Alarifi, I. M., *et al.*, Investigation of Thermal Runaway in Lithium-Ion Batteries, *Journal of Energy Storage, 8* (2016), May, pp. 161-169

Paper submitted: December 3, 2024
Paper revised: February 11, 2025
Paper accepted: May 4, 2025