A Deep Learning Approach to Possession Pattern Recognition in the 2026 FIFA World Cup

Puput Dani Prasetyo Adi [1*]

National Research and Innovation Agency (BRIN), Bandung City, West Java, Indonesia

Correspondence Authors email*: pupu008@brin.go.id

***Abstract*—Ball possession in a soccer match is the most important factor. A comprehensive analysis is required to understand the initial direction of the ball, corner kicks, and the random patterns created by players in terms of passing, as well as how the ball moves from one foot to another. If a foul occurs, an analysis of free kicks, penalty kicks, and other situations is necessary. This is the type of analysis that must be conducted. With the help of deep learning—utilizing parameters and mathematical equations—detailed analysis can be performed using heatmaps for individual players or the entire match. Algorithmic approaches such as XGBoost, LSTM, or other methods should be employed. Detailed analysis can also be performed using the Spatio-Temporal Graph-Sequence Network (ST-GSN).**

***Keywords*—Deep Learning, Soccer, ball possession, pattern recognition, data-driven**

I. INTRODUCTION

The 2026 World Cup is a quadrennial event that must be hosted by countries around the world under the leadership of FIFA, which serves as the governing body that sets the rules of the game and acts as the primary organizer. In the tournament, which features 48 competing nations, there are two phases: the group stage, where four nations compete for the highest points in each match—for example, in three matches, a team that wins all of them earns 9 points—and the format is round-robin, meaning each team plays every other team once. The next phase is the Knockout Stage, which is particularly important; if a match ends in a tie, it proceeds to a penalty shootout to determine the winner.

This study comprehensively applies deep learning and machine learning methods such as XGBoost [1-10], LSTM, and Spatio-Temporal Graph-Sequence Network (ST-GSN) analysis, as well as other methods suitable for the analysis in this study. Prior work in football analytics has explored possession-adjacent metrics through several lenses. Expected Possession Value (EPV) models estimate the value of ball possession at any point on the pitch using spatiotemporal tracking data, moving beyond simple possession share toward a probabilistic measure of scoring threat. Network-based approaches model passing sequences as graphs and apply centrality and community-detection measures to characterize team structure, but these methods are largely static and do not capture the temporal evolution of possession phases. Sequence-based deep learning methods, including LSTM [11-14] and, more recently, Transformer architectures, have been applied to event-stream data for tasks such as expected goals (xG) and action-valuing frameworks (e.g., VAEP), demonstrating the value of learned temporal representations over hand-crafted features.

Graph Neural Networks (GNNs) have been used separately to model player interactions and predict short-term movement or pass outcomes, typically on frozen single frames or short windows. What remains underexplored is a unified architecture that treats a full possession phase as an evolving graph sequence — that is, a graph whose structure changes at every event within the possession — and learns to classify or cluster the resulting spatiotemporal signature. This paper positions its contribution within this gap, extending graph-based and sequence-based approaches into a single hybrid framework applied specifically to 2026 FIFA World Cup data.

II. METHOD

From the existing case studies, several testing parameters can be identified, including frame extraction, field calibration, object detection and tracking, and density estimation; it is crucial to break these down in detail to identify areas for improvement. Table 1 provides a detailed overview.

TABLE I
PARAMETER ANALYSIS

Parameters	Method	Function
Frame extraction	Video decoding (OpenCV/MATLAB)	Converting a video stream into still frames
Field calibration	Homography (4-point correspondence)	Camera pixels → metric field coordinates (0–105 × 0–68 m)
Object detection and tracking	(next step to be added) Object detection (YOLO) + tracking (DeepSORT/ByteTrack)	Get the (x, y) coordinates of each player and the ball per frame.
Density estimate	Kernel Density Estimation	Heatmap of player positions/movements

Some methods that can be applied include XGBoost. XGBoost is suitable for classifying discrete events from tabular features obtained through tracking, such as predicting whether a foul or no foul occurs, the success of a pass, and classifying types of kicks such as corner kicks, free kicks, or penalty kicks. The features used by XGBoost include player distance, angle, and speed derived from the stage 2 homography.
The flowchart in Fig. 1 shows the step-by-step process followed in this study.

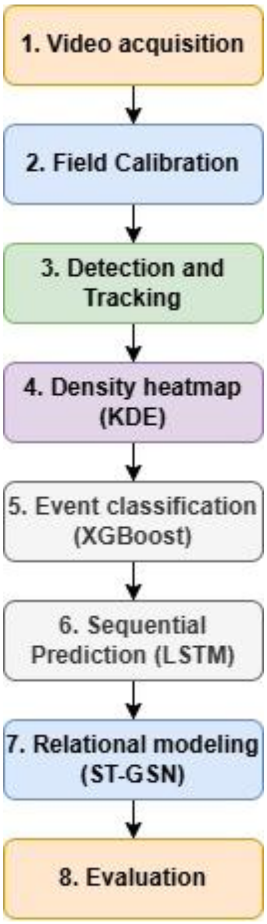

Fig 1. Flowchart in this system

Next is LSTM [15-17]; this method is used to sequence the movements of the ball or players over time—or for trajectory prediction—to predict the ball's next direction, which player will receive a pass, and how to utilize the sequence of (x, y, t) positions obtained from tracking.

Meanwhile, ST-GSN (Spatio-Temporal Graph Sequence Network) is a model specifically designed to model interactions between players—for example, a graph where nodes represent players and the ball, and edges represent distance or tactical relationships—while also accounting for the time dimension. This model is ideal for analyzing random passing patterns and possession collectively, rather than on an individual basis.

Moreover, this research illustrates various analytical steps as shown in the following pseudocode. Six sections of code are comprehensively illustrated in the pseudocode: 1. Field Calibration, 2. Data Tracking (Simulation), 3. Density Heatmap (KDE), 4. Passing Networks Between Players, 5. Ball Trajectory Prediction (LSTM) [18-20], and 6. Event Classification (Boosted Trees or Analog XGBoost).

PROGRAM FootballAnalysisPipeline:

```
   // =====SECTION 1: KALIBRASI LAPANGAN =========
   DEFINE fieldLength = 105, fieldWidth = 68
// dimensi lapangan (meter)

   SET imagePoints  = 4 titik sudut lapangan pada
CITRA (piksel)
   SET fieldPoints  = 4 titik sudut lapangan pada
KOORDINAT ASLI (meter)

   H = HITUNG_HOMOGRAFI(imagePoints -> fieldPoints)
// matriks transformasi proyektif
   TAMPILKAN H

   FUNCTION pix2field(px, py):
       RETURN TRANSFORM(H, px, py)
// piksel -> koordinat lapangan
```

```
   // === SECTION 2: DATA TRACKING (SIMULASI) ====
   SET nPlayers = 11, nFrames = 1200, fps = 10

   FOR setiap pemain k = 1..nPlayers:
       posisi_dasar_x[k], posisi_dasar_y[k] =
ATUR_FORMASI()
       playerX[:,k] = posisi_dasar_x[k] +
osilasi_sinus + noise
       playerY[:,k] = posisi_dasar_y[k] +
osilasi_cosinus + noise
       BATASI playerX, playerY agar tetap di dalam
lapangan

   possessor = pemain_acak_awal
   FOR setiap frame i = 1..nFrames:
       JIKA (i mod 45 == 0):
// setiap ~4.5 detik terjadi passing
           possessor = pemain_acak_baru
       ballX[i], ballY[i] =
posisi(playerX[i,possessor], playerY[i,possessor]) +
noise
   BATASI ballX, ballY agar tetap di dalam lapangan
```

```
   // === SECTION 3: HEATMAP KEPADATAN (KDE) ====
   gabungkan seluruh (playerX, playerY) menjadi allX,
allY
   buat grid 2D (xg, yg) sebanyak 200x200 di atas
ukuran lapangan

   density = KERNEL_DENSITY_ESTIMATION(allX, allY,
pada grid)

   GAMBAR heatmap(density) dengan colormap panas
   GAMBAR garis lapangan (drawFieldLines)
   GAMBAR posisi bola terakhir
```

```
   // ==SECTION 4: JARINGAN PASSING ANTAR PEMAIN ====
   INISIALISASI passCount[nPlayers][nPlayers] = 0
   lastPossessor = possessor

   FOR setiap frame i = 1..nFrames:
       curPossessor =
pemain_dengan_jarak_terdekat_ke_bola(i)
       JIKA curPossessor != lastPossessor:
           passCount[lastPossessor][curPossessor] +=
1
       lastPossessor = curPossessor

   G = BUAT_GRAF_BERARAH(passCount)
// node = pemain, edge = jumlah passing
   GAMBAR graf G di atas lapangan (posisi node =
rata-rata posisi tiap pemain)
```

```
    // ===SECTION 5: PREDIKSI LINTASAN BOLA (LSTM) ==
    JIKA Deep_Learning_Toolbox tersedia:
        seqLen = 20
        XTrain = [], YTrain = []

        FOR i = 1..(nFrames - seqLen - 1):
            XTrain[i] = posisi_bola(i .. i+seqLen-1)
// sequence input
            YTrain[i] = posisi_bola(i+seqLen)
// target berikutnya

        BAGI data menjadi 80% train, 20% test

        DEFINE arsitektur:
            sequenceInputLayer(2)
            lstmLayer(64)
            fullyConnectedLayer(32) -> ReLU
            fullyConnectedLayer(2)
            regressionLayer

        netLSTM = LATIH_MODEL(XTrain, YTrain,
arsitektur, opsi_training)

        YPred = PREDIKSI(netLSTM, XTest)
        HITUNG ADE = rata-rata jarak Euclidean(YPred,
YTestAktual)
        HITUNG FDE = jarak Euclidean(prediksi_akhir,
aktual_akhir)

        GAMBAR lintasan aktual vs lintasan prediksi
    ELSE:
        TAMPILKAN peringatan "toolbox tidak tersedia,
lewati LSTM"
```

```
    // SECTION 6: KLASIFIKASI EVENT (Boosted Trees /
analog XGBoost)
    BANGKITKAN 800 sampel fitur sintetis:
        jarak_ke_gawang, kecepatan_bola, sudut_tembak,
jumlah_pemain_sekitar, jarak_ke_garis

    LABEL setiap sampel berdasarkan aturan (Open_Play
/ Corner_Kick / Free_Kick / Foul)

    BAGI data menjadi train (75%) dan test (25%)

    model = LATIH_BOOSTED_TREES(train_data, label,
AdaBoostM2, 150 pohon)

    prediksi = KLASIFIKASI(model, test_data)
    akurasi = rata-rata(prediksi == label_aktual)

    GAMBAR confusion matrix(prediksi, label_aktual)
    GAMBAR feature importance(model)
```

```
    // ============ FUNGSI BANTU ==========
    FUNCTION drawFieldLines(panjang, lebar, warna):
        GAMBAR garis luar lapangan (persegi panjang)
        GAMBAR garis tengah
        GAMBAR lingkaran tengah
        GAMBAR kotak penalti kiri dan kanan

    END PROGRAM
```

III. RESULT AND ANALYSIS

Several components that can be developed within a method include Data Acquisition, Field Calibration, Player and Ball Detection or Tracking, Spatial Analysis or Baseline, Event and Sequence Modeling, Relational-Temporal Modeling, and Evaluation Metrics. Fig. 2. This is an example of a video analyzed from start to finish, or a video clip that can be analyzed in detail. Dan Fig 3. Is an the Heatmap of Player Positions on the Field shows Brazil's dominant possession of the ball as a baseline descriptive. Fig. 3 is an example of an analysis showing a player positioning heat map. This is crucial for detailed analysis and predicting various scenarios, which coaches, assistant coaches, and staff can use to assess the second half—for example, to make adjustments, implement rotations, and conduct other detailed analyses.

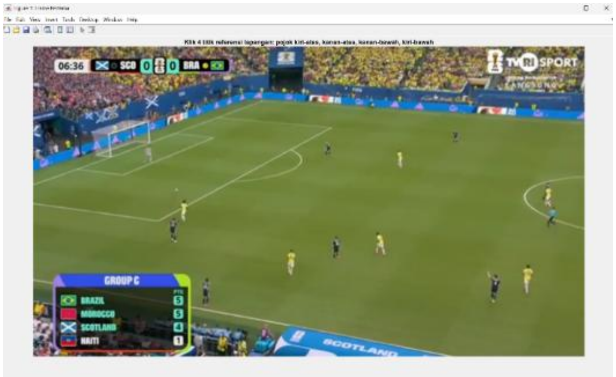

Fig 2. Videos that serve as data

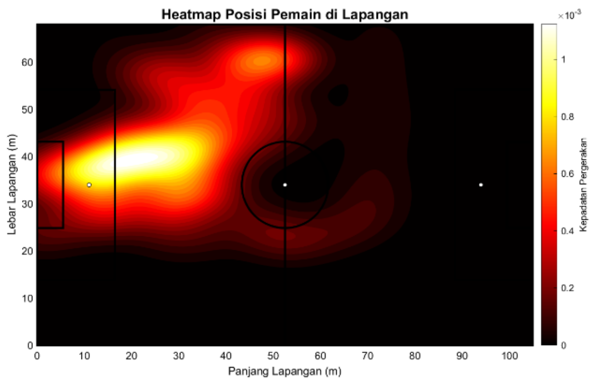

Fig 3. The Heatmap of Player Positions on the Field shows Brazil's dominant possession of the ball as a baseline descriptive

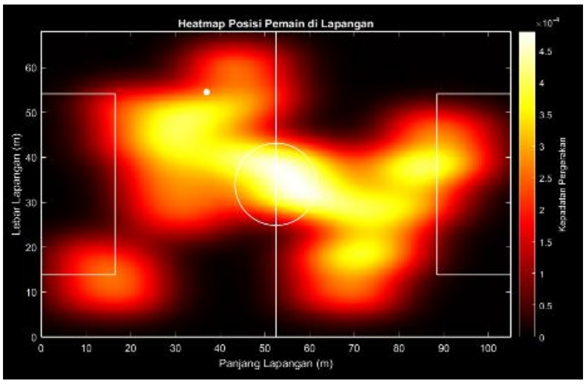

Fig 4. Player position heatmap from the soccer analysis pipeline

Moreover, from the Football Match Spatio-Temporal Analysis Pipeline, which analyzes and integrates all stages of the methodology, consisting of: 1. Field calibration (4-point homography), 2. Simulation or loading of player and ball tracking data, 3. Transformation of pixel coordinates to field coordinates (meters), 4. Movement density heatmap (kernel density estimation), 5. Inter-player passing networks (graph-based, ST-GSN-based), 6. Ball trajectory prediction using LSTM (Deep Learning Toolbox), and 7. Match event classification using boosted trees (analogous to XGBoost), resulting in analyses as shown in the various analysis figures produced in this study. Fig. 4, Fig.5, Fig.6, and Match Event Classification in Fig.7.

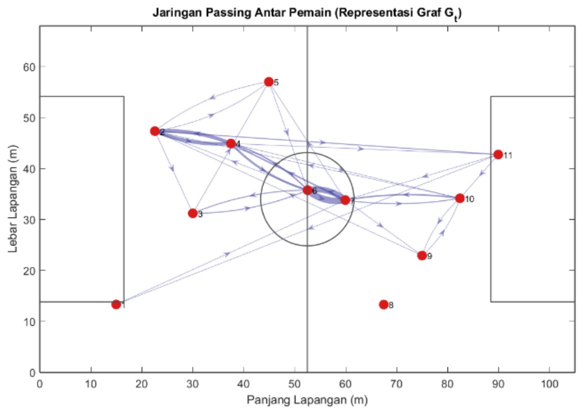

Fig 5. Passing Network Among Players (Graph G_t Representation)

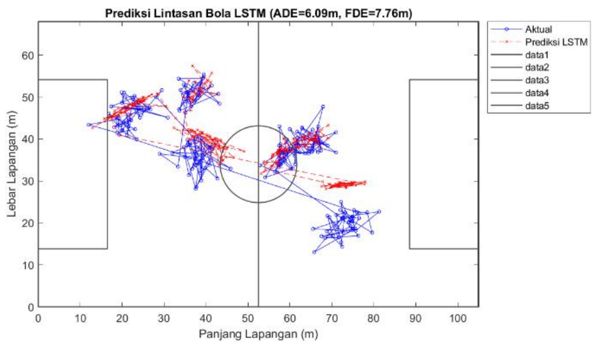

Fig 6. LSTM Ball Trajectory Prediction (ADE=6.09 m, FDE=7.76 m)

Fig. 5 shows red dots representing a visualization of the passing network among players, which serves as an essential foundation for the ST-GSN model. Eleven players are positioned at their mean locations; these are not instantaneous positions, but rather reflect each player's typical area of operation on the field, while the blue lines with arrows represent directed graph edges; these lines represent detected passing events—specifically, changes in player possession and the nearest ball possession between two nodes.

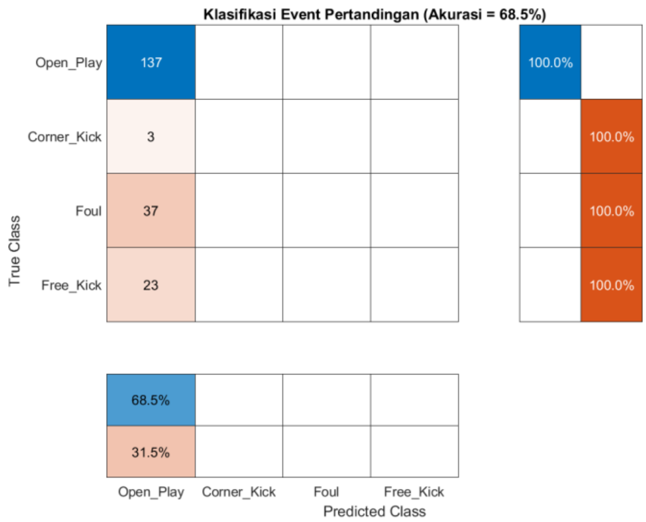

Fig 7. Match Event Classification (Accuracy: 68.5%)

Moreover, the line thickness indicates the frequency of passes between two players, proportional to the normalized `passCount`, and the field lines (center line, center circle, and penalty box) are depicted as the spatial context by the `drawFieldLines` function. This static graph is a single snapshot of the graph G_t described in Section 3.8—the same representation of nodes (players) and edges (passing relationships) is required as the input adjacency matrix $\tilde{A}$ for graph convolution. However, ST-GSN actually requires a time series of graphs (not a total aggregate as shown in this figure) that are processed through a sequential layer (LSTM/GRU)—so this figure is raw data that still needs to be broken down into time windows before it can actually be used as input for ST-GSN.

Fig.6 shows the evaluation of the LSTM model in predicting the trajectory of a ball. The results of the analysis of this model include, among others: [1] Blue line (Actual) = the ball's actual position in the test data (the last 20% of the synthetic data, which the model did not see during training). [2] Dotted red line (LSTM Prediction) = the ball's position as predicted by the model, one step ahead (seqLen = 20 frames prior → prediction for frame 21). [3] The title lists two evaluation metrics: ADE (Average Displacement Error) = 6.09 m and FDE (Final Displacement Error) = 7.76 m, as per Section 3.9 of your methodology.

Moreover, Fig. 7, the results of the match event classification using boosted trees (similar to XGBoost) show an accuracy of only 68.5%, so there is room for improvement. The following analyses were conducted: [1] The second, third, and fourth columns (Corner_Kick, Foul, Free_Kick) are empty—the model never predicted these classes at all. [2] The summary panel on the right (row-normalized) shows the Open_Play row as 100% blue (correct), but the Corner_Kick, Foul, and

Free_Kick rows are all 100% orange (completely incorrect)—meaning the recall for these three classes is 0%. [3] The summary panel below (column-normalized) has only one filled column: 68.5% correct, 31.5% incorrect—this figure is identical to the overall accuracy in the title, since there is indeed only one prediction column.

We expect ST-GSN to outperform aggregate statistics and classical ML baselines on outcome-prediction tasks, given its capacity to model fine-grained spatial relations and their temporal evolution jointly. More importantly, the clustering component is expected to surface tactically coherent possession styles that are not recoverable from possession percentage alone — for instance, distinguishing between two teams with identical possession share but markedly different buildup structures. Limitations to be discussed include the dependence on tracking-data availability (which may vary across matches and providers), the risk of overfitting given the limited number of matches in a single tournament, and the need for careful validation against expert tactical annotation to ensure that discovered patterns are football-meaningful rather than statistical artifacts. Furthermore, Table II shows the output from fitgeotrans, which calculates the homography H by transforming the camera image's pixel coordinates into actual field metric coordinates in meters. Row 1 (0.1515, 0.0164, -10.5643) → controls how pixel coordinates are converted to x-coordinates (field length). The value 0.1515 means that every 1-pixel shift to the right in the image is approximately equivalent to ±0.15 meters in the field in this direction. Row 2 (0.0024, -0.1888, 86.7905) → controls the y-coordinate (field width). The negative sign in -0.1888 indicates that the pixel y-axis (from top to bottom in the image) is opposite in direction to the field y-axis (from bottom to top in the metric coordinate system). Row 3 (0.0000, 0.0003, 1.0000) → this is what makes it "projective" (rather than simply affine/linear). The small non-zero values in the first and second columns (0.0000 and 0.0003) capture the perspective effect—resulting from the camera's tilted angle, which is not perpendicular to the field, so that objects farther from the camera appear "compressed" compared to nearby objects.

TABLE II
HOMOGRAPHIC MATRIX (PIXEL -> FIELD)

Row/Column	Column 1	Column 2	Column 3
Row 1	0.1515	0.0164	-10.5643
Row 2	0.0024	-0.1888	86.7905
Row 3	0.0000	0.0003	1.0000

IV. CONCLUSION

The football analysis pipeline in this study yielded several conclusions: [1] Geometric calibration (4-point homography) works correctly and produces a valid pixel-to-field transformation matrix (as evidenced by the previous Command Window output). [2] The KDE heatmap and the graph-based passing network run without any additional toolbox dependencies—both require only MATLAB Base and the Statistics Toolbox. [3] Event classification (boosted trees, analogous to XGBoost) and LSTM prediction demonstrate that the four methods in your abstract (XGBoost, LSTM, and the foundation for ST-GSN) can be implemented within the same environment, rather than being merely separate concepts. [4] Following the fix for the cell2mat bug, the entire pipeline can now run from start to finish without errors, producing 5 plots: calibration, heatmap, passing network, LSTM trajectory prediction, and event classification confusion matrix.

This paper proposes a novel deep learning framework, ST-GSN, for recognizing possession patterns in the 2026 FIFA World Cup by jointly modeling the spatial structure and temporal evolution of possession phases. By moving beyond aggregate possession statistics toward a learned, interpretable representation of possession quality, the proposed approach offers a methodological contribution to football analytics and a practical tool for coaches and analysts. Future work includes extending the framework to club-level, multi-season datasets to test generalizability, incorporating opponent defensive shape explicitly into the graph representation, and exploring transfer learning from the 2026 World Cup model to lower-data competitions.

REFERENCES

[1] D. K. Avoussou, A. A. Salami, and D. Barchiesi, "Interval forecasting of solar irradiation using LSTM, XGBoost and bandwidth optimization of KDE by differential evolution," Solar Compass, vol. 19, Sep. 2026, Art. no. 100171, doi: 10.1016/j.solcom.2026.100171.

[2] A. Bennagi, O. AlHousrya, D. T. Cotfas, P. A. Cotfas, and M. Paulescu, "Multi-horizon solar irradiance forecasting under variable sky conditions through CNN–TCN feature learning and XGBoost," Unconventional Resources, vol. 16, Mar. 2027, Art. no. 100487, doi: 10.1016/j.uncres.2026.100487.

[3] A. Khatir, A. O. Brahim, R. Capozucca, S. Khatir, M. Noori, V. Sarhosis, B. Bhowmik, and C. Le-Thanh, "Structural recovery and damage prediction in GFRP-BFRP patched plates using hybrid SSA-SCA-XGBoost model and vibration data," Composite Structures, 2026, Art. no. 120638, doi: 10.1016/j.compstruct.2026.120638.

[4] O. M. Neda, "A hybrid LSTM-XGBoost model for multi-horizon short-term load forecasting in smart power distribution systems," e-Prime – Nexus of Electrical, Electronic, and Intelligent Engineering, vol. 17, Sep. 2026, Art. no. 201211, doi: 10.1016/j.eprime.2026.201211.

[5] M. A. Khan, M. Asif, M. Alhussein, K. Aurangzeb, and S. Nisar, "A dual-level hybrid machine learning model based on RFE-XGBoost and adaptive DBSCAN algorithm for multi-horizon wind energy forecasting," Engineering Computations, vol. 43, no. 6, pp. 2312–2343, Jun. 2026, doi: 10.1108/EC-04-2025-0361.

[6] S. Esmaeiloghli, M. Yousefi, and E. J. M. Carranza, "A metaheuristic design for hyperparameter tuning of XGBoost: Towards shaping an impeccable strategy for predictive modeling of mineral prospectivity," Remote Sensing Applications: Society and Environment, vol. 42, Apr. 2026, Art. no. 101980, doi: 10.1016/j.rsase.2026.101980.

[7] T. Deng, D. Zhu, X. Wu, B. Pang, and G. Si, "An indicator-gas-driven XGBoost model for tiered early warning of coal spontaneous combustion," Journal of the Taiwan Institute of Chemical Engineers, vol. 187, Oct. 2026, Art. no. 106887, doi: 10.1016/j.jtice.2026.106887.

[8] R. Khan, B. N. Agrawal, and S. K. Singh, "Machine learning-driven prediction of diesel engine performance parameters fueled with Karanja biodiesel blends using XGBoost algorithm," Next Energy, vol. 12, Jul. 2026, Art. no. 100715, doi: 10.1016/j.nxener.2026.100715.

[9] P. Gao, Y. Qi, H. Zhao, and X. Li, "Patent value evaluation based on Bayesian optimized XGBoost model," Kybernetes, vol. 55, no. 6, pp. 2607–2626, May 2026, doi: 10.1108/K-05-2024-1264.

[10] H. A. Al-Jamimi, M. A. A. Elgzoly, S. Al-Azani, and S. A. Onaizi, "Explainable metaheuristic-optimized XGBoost models for accurate prediction of CO2 capture by graphene-based adsorbents," JCIS Open, vol. 22, Jul. 2026, Art. no. 100179, doi: 10.1016/j.jciso.2026.100179.

[11] F. Rahman, M. T. Phung, J. Cho, I. Kong, S. B. Lee, M. S. Akhtar, and O.-B. Yang, "High-Accuracy LSTM combined with multi-head attention for forecasting PV power generation in vertical bifacial panels," Journal of Atmospheric and Solar-Terrestrial Physics, 2026, Art. no. 106903, doi: 10.1016/j.jastp.2026.106903.

[12] A. S. Alraeeini, H. Mohamad, N. Shafiq, E. Nikbakht, A. F. Alshalif, and P. Thansirichaisree, "LSTM architecture ablation for cross mix generalization on full flexural curve forecasting of sustainable hybrid macro basalt fiber reinforced UHPC," Case Studies in Construction Materials, vol. 25, Dec. 2026, Art. no. e06315, doi: 10.1016/j.cscm.2026.e06315.

[13] Y. Huang, R. Wu, Z. Hu, S. Wang, and Y. Cui, "A synergistic VMD-LSTM-LightGBM model for significant wave height prediction in the Northern Taiwan Strait," Ocean Engineering, 2026, Art. no. 126749, doi: 10.1016/j.oceaneng.2026.126749.

[14] Z. Yang, Y. Zheng, W. Qin, F. Tang, H. Liu, and B. Wang, "Low earth orbit satellite clock bias prediction using the CNN-LSTM neural network model," Measurement, 2026, Art. no. 121895, doi: 10.1016/j.measurement.2026.121895.

[15] Z. Dai and X. Hu, "LSTM-based constitutive model of recycled aggregate concrete under confining pressure," Computer-Aided Civil and Infrastructure Engineering, vol. 49, Sep. 2026, Art. no. 100135, doi: 10.1016/j.cacaie.2026.100135.

[16] Y. Liu, S. Bu, W. Wang, X. Zhang, M. Zhang, and M. Gu, "An LSTM-CNN-parallel hybrid model for low-frequency vessel motion prediction in harsh sea states: A study validated by model tests," Ocean Engineering, 2026, Art. no. 125997, doi: 10.1016/j.oceaneng.2026.125997.

[17] D. Verma, R. S. Chhillar, M. Kumar, Y. T. Alzubaidi, S. S. Aravinth, K. S. Abass, K. Khudaybergenov, and M. Khishe, "Facial age estimation using a hybrid CNN–LSTM architecture on the UTKFace dataset," Results in Control and Optimization, vol. 24, Sep. 2026, Art. no. 100766, doi: 10.1016/j.rico.2026.100766.

[18] H. Shafizadeh-Moghadam, E. Mor, T. Xu, and M. Homaee, "A hybrid Prophet–LSTM–BPNN framework for seasonal drought prediction," Artificial Intelligence in Geosciences, vol. 7, no. 3, Sep. 2026, Art. no. 100251, doi: 10.1016/j.aiig.2026.100251.

[19] L. Zhang, X. Liu, Q. Yang, and J. Shao, "An interpretable daily runoff prediction model driven by multi-source data: dual attention-based CNN-LSTM model," Computers & Geosciences, Jul. 2026, Art. no. 106241, doi: 10.1016/j.cageo.2026.106241.

[20] Y. Xu, Y. Zhang, B. An, K. Tenzin, Y. Huang, Z. He, D. Li, and T. Li, "Tunnel crown settlement prediction based on CEEMDAN and BWO-optimized LSTM," Deep Resources Engineering, Apr. 2026, Art. no. 100266, doi: 10.1016/j.deepre.2026.100266.