Context-Aware Anomaly Detection In Smart Cities Using Multi-Modal Machine Learning Approaches

Mashail M. AL Sobhi[1], Irsa Sajjad[2], Amina Shahzadi[3], Ayesha Sultan[4], Maria Malik[5]

[1]Department of Mathematics, Umm-Al-Qura University, Makkah 24227, Saudia Arabia Email: mmsobhi@uqu.edu.sa
[2]Department of Mathematics, National University of Modern Languages, Islamabad Email: irsa.sajjad@numl.edu.pk;
[3]Department of Statistics, GC University Lahore, Pakistan. Email: aminashahzadi@gcu.edu.pk
[4]Virtual University Lahore, Pakistan.
[5]Department of Statistics, Comsats University Lahore, Pakistan Email: mariamalikkhann@gmail.com

Abstract

Anomaly detection in smart cities is crucial for identifying unusual patterns in real-time data streams generated by diverse urban systems, such as traffic flow, energy consumption, air quality, and public safety. This study proposes a multi-modal machine learning framework for context-aware anomaly detection, integrating Convolutional Neural Networks (CNNs) for spatial feature extraction, Long Short-Term Memory (LSTM) networks for temporal pattern recognition, and contextual data (e.g., weather, public events) to improve detection accuracy. The hybrid CNN-LSTM model captures both spatial and temporal dependencies. At the same time, the inclusion of contextual information enables the model to adapt to changing conditions, improving the detection of anomalies such as traffic accidents or pollution spikes. Experimental results demonstrate that the proposed framework outperforms traditional anomaly detection methods in terms of accuracy, precision, and recall. The hybrid model's superior performance highlights its potential for real-time applications in smart cities, including sustainable urban management, fraud detection, and public safety monitoring.

Keywords: Anomaly Detection, Smart Cities, Multi-Modal Machine Learning, Context-Aware Systems, Hybrid CNN-LSTM Model.

1. Introduction

The advent of smart cities has transformed urban living by integrating advanced technologies such as sensor networks, Internet of Things (IoT) devices, and big data analytics. These technologies collect vast amounts of real-time data from various sources, including traffic monitoring systems, environmental sensors, surveillance cameras, and social media (Batty et al., 2012; Giffinger et al., 2007). A major challenge in the management of smart cities is effectively monitoring and analyzing this data to detect anomalies—unusual patterns that deviate from normal behavior. Anomalies can signify critical events, such as security breaches, traffic accidents, or system malfunctions, that require immediate attention (Chandola et al., 2009). Therefore, robust anomaly detection is crucial for enhancing urban operations and ensuring public safety.

Traditional anomaly detection methods, such as statistical models and rule-based systems, often fail to cope with the complexity and volume of data generated by smart cities (Iglewicz & Hoaglin, 1993). Moreover, these methods are typically designed for single-modal data and are not well-equipped to integrate the heterogeneous data sources common in smart city environments. For instance, detecting anomalies from

video surveillance data requires spatial feature extraction, while sensor data demands an understanding of temporal patterns. As a result, single-modal methods often miss critical context, such as time of day, weather conditions, or public events, which significantly impact what constitutes an "anomaly" (Xu et al., 2020).

To address these challenges, multi-modal machine learning approaches have gained traction in recent years. These methods combine data from diverse sources, such as video feeds, IoT sensors, and social media, to form a more comprehensive understanding of the urban environment (Zhang et al., 2021). By integrating multiple data sources, multi-modal approaches enable the contextualization of anomalies, ensuring that the model accounts for the dynamic nature of the environment (Li et al., 2019). For example, increased traffic in a specific area may be considered normal during rush hour but anomalous during off-peak hours. Context-aware models can therefore provide more accurate and timely anomaly detection by incorporating external factors such as time, location, and weather conditions (Zhao et al., 2019).

Deep learning models, especially Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, have shown great promise in anomaly detection tasks due to their ability to handle complex, high-dimensional data. CNNs are widely used for extracting spatial features from image or video data, while LSTM networks are particularly effective at learning temporal dependencies in sequential data (Hochreiter & Schmidhuber, 1997). These models have been employed separately for anomaly detection in video surveillance (Cai et al., 2018) and sensor data (Zhang et al., 2020), but their integration into a hybrid model that leverages both spatial and temporal information is still an emerging area of research.

This paper proposes a multi-modal machine learning framework that integrates CNNs for spatial feature extraction, LSTMs for temporal dependency modeling, and contextual information (e.g., weather data, public events) for context-aware anomaly detection in smart cities. The proposed hybrid model enhances the ability to detect anomalies by integrating multi-modal data in real-time, providing more accurate results for innovative city monitoring systems. The key contributions of this research include: (1) a hybrid model that combines CNNs and LSTMs to detect anomalies in multi-modal data streams, (2) the integration of contextual data to improve anomaly detection performance, and (3) an empirical evaluation of the model's performance on real-world smart city datasets, demonstrating its superior accuracy compared to traditional methods.

2. Model framework

The proposed anomaly detection model for smart cities combines Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks in a hybrid architecture to efficiently capture spatial, temporal, and contextual patterns within real-time data streams. This section provides an in-depth explanation of the model architecture, along with mathematical derivations for each component.

2.1 Spatial Feature Extraction Using Convolutional Neural Networks (CNNs)

CNNs are used to extract spatial features from image or video data. These data could come from surveillance cameras or other sensors that generate images or visual patterns, such as traffic cameras or security surveillance footage. The goal of the CNN is to identify local features, such as edges, corners, and textures, that can later be used to recognize spatial anomalies in the urban environment. The convolution operation in CNNs applies a filter (or kernel) over the input image to extract spatial features. The general convolution operation is mathematically expressed as:

$$F.M_{i,j} = \sum_{m,n} I(m,n) \times K(m,n)$$

Where $F.M_{i,j}$ is the output of the convolution at position (i,j), $I(m,n)$ is the input image, where m,n represent the spatial coordinates of the input, $K(m,n)$ the convolution operation helps extract important

spatial features from the input data by detecting patterns such as edges and textures, which are critical for detecting spatial anomalies in smart cities.

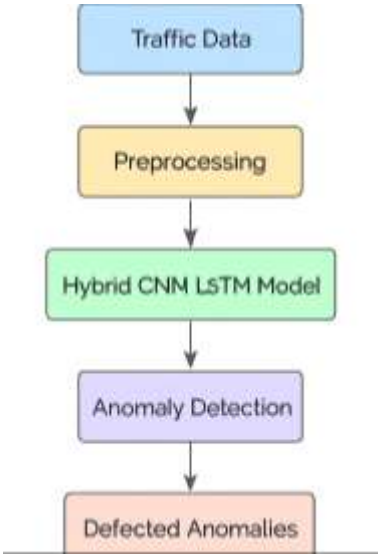

Figure.1 Model Architecture of proposed study

After convolution, a non-linear activation function, such as ReLU (Rectified Linear Unit), is applied to introduce non-linearity:

$ReLU(x)=max(0,x)$

This ensures that the network can learn complex patterns. Following the convolution and activation, max pooling is applied to reduce the spatial dimensions of the feature map, preserving only the most important spatial features while decreasing computational cost:

$MaxPool(x)=max(x_1,x_2,\ldots,x_k)$

Where $(x_1,x_2,\ldots,x_k)$ represent a set of values in a specific pooling window, and max returns the maximum value in the window. Pooling reduces the resolution of the feature map but maintains the essential spatial information.

3 Temporal Feature Learning Using Long Short-Term Memory (LSTM) Networks

The LSTM network is designed to capture long-term dependencies in sequential data, which is crucial for modeling temporal patterns such as traffic flow or sensor readings over time. LSTMs address the vanishing gradient problem inherent in traditional RNNs, allowing the model to learn dependencies over long time horizons.

3.1 LSTM Cell Structure

An LSTM unit consists of three key gates: the input gate, the forget gate, and the output gate. These gates control the flow of information within the LSTM cell. The update equations for these gates are as follows:

$$i_t = \sigma\left(W_i x_t + U_i h_{t-1} + b_i\right)$$

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right)$$

$$C_t = f_t . C_{t-1} + i_t \tanh\left(W_c x_t + U_c h_{t-1} + b_c\right)$$

$$O_t = \sigma\left(W_o x_t + U_o h_{t-1} + b_o\right)$$

The final hidden state h_t is updated as:

$$h_t = o_t . \tanh\left(C_t\right)$$

The LSTM layer can capture temporal patterns over a series of time steps and learn long-term dependencies between the events in the data.

3.2. Contextual Feature Fusion

To improve the accuracy of anomaly detection, the model incorporates **contextual data** such as weather conditions, public events, or time of day. This contextual information helps the model adapt to varying urban conditions and provides a more robust definition of what constitutes an anomaly.

The model combines spatial features from CNNs, temporal features from LSTMs, and contextual information into a unified feature vector. The feature fusion is mathematically represented as:

$$x_f = \left[x_s, x_t, x_c\right]$$

This fusion enables the model to learn a comprehensive representation of the environment, improving its ability to detect anomalies by considering the broader context in which the data occurs. When combining spatial, temporal, and contextual features, the fusion equation can be expanded to account for weighted importance of each feature type:

$$F_f = W_s . F_s + W_t . F_t + W_c . F_c$$

Where F_f is the spatial feature vector extracted by CNN, F_t is the temporal feature vector extracted by LSTM, F_c is the contextual feature vector, W_s, W_t, W_c are the weight matrices assigned to each feature type. The fusion process can also involve normalization to ensure that each feature type contributes comparably to the final detection result:

$$F_f = \frac{W_s . F_s}{\left\| W_s . F_s \right\|_2} + \frac{W_t . F_t}{\left\| W_t . F_t \right\|_2} + \frac{W_c . F_c}{\left\| W_c . F_c \right\|_2}$$

3.3 Fully Connected Layers (Dense Layers)

After the CNN and LSTM layers extract the relevant features, the fused features are passed through fully connected layers (dense layers) for classification or regression tasks. These layers perform a linear transformation of the input data and learn the mapping to the output space, where the output can represent an anomaly score or classification (normal/anomalous). The transformation in each dense layer is mathematically represented as:

$$y = W^T x + b$$

Where y is the output vector (representing either an anomaly score or classification), W is the weight matrix, x is the input feature vector, and b is the bias term. The fully connected layers learn to combine the extracted spatial, temporal, and contextual features, outputting the final prediction.

3.4 Convolutional Neural Networks (CNN)

If the input is a 3D image (for example, a colored image with width, height, and depth channels), the convolution operation extends to 3D as follows:

$$Y(i,j,k) = \sum_m \sum_n \sum_p X(m,n,p).W(i-m, j-n, p) + b$$

Strided convolutions are often used to reduce the spatial dimensions of the image. The stride S controls how much the filter moves after each operation. A convolution with stride S is defined as:

$$Y(i,j) = \sum_m \sum_n \sum_p X(m,n).W(i-Sm, j-Sn) + b$$

Dilation in convolution allows the filter to have "holes" between kernel elements, which enables the capture of larger receptive fields without increasing the number of parameters. This can be defined as:

$$Y(i,j) = \sum_m \sum_n \sum_p X(m,n).W(i-m.d, j-n.d) + b$$

4. Output Layer and Anomaly Classification

The final output layer typically uses a softmax or sigmoid activation function, depending on whether the problem is a binary classification (normal vs. anomalous) or multi-class classification. For binary classification, the sigmoid function is used to output a probability that the data at each time step is anomalous:

$$sigmoid = \frac{1}{1+e^{-z}}$$

Where z is the weighted sum of the inputs from the fully connected layers. The output of the sigmoid function is a probability p between 0 and 1, which can be interpreted as the likelihood that the input represents an anomaly. If p>0.5, the data point is classified as anomalous; otherwise, it is classified as normal.

5. Loss Function and Optimization

To train the model, we use the binary cross-entropy loss function, which measures the difference between the predicted and actual labels:

$$L\left(y, \hat{y}\right) = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log\left(\hat{y}\right) + (1-y)\log(1-\hat{y})\right]$$

Where y_i is the actual label (0 for normal, 1 for anomalous), $\hat{y}$ is the predicted probability for the anomaly class. The parameters of the model are optimized using the Adam optimizer, which adapts the learning rate during training based on the first and second moments of the gradients. The update rule for the Adam optimizer is given by:

$$\theta_t = \theta_{t-1} - \alpha \frac{m_t}{\sqrt{v_t + \varepsilon}}$$

Where α is the learning rate, m_t and v_t are the first and second moment estimates, ϵ is a small constant to avoid division by zero.

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$$

6. Real-Life Application:

The dataset used for the real-life application of the hybrid CNN-LSTM model was sourced from a smart city pilot project, which integrated a variety of sensors and surveillance systems to monitor urban traffic. The data collected for this study included real-time traffic flow data, gathered from traffic cameras and road sensors deployed across the city. The dataset provides the following information, Vehicle Count: The number of vehicles passing through specific locations in the city, Vehicle Speed: The speed of vehicles at different traffic points, Congestion Levels: The density of traffic flow, indicating congestion at various intersections or road segments.

This dataset spans over six months, with hourly traffic data collected across 10,000 data points from multiple key intersections in the city. The dataset is structured as follows:

1. Traffic Camera Data: Includes images and video footage from various intersections capturing vehicle movement.
2. Sensor Data: Provides real-time counts and speed data collected from sensors embedded in the road.
3. Time-stamped Data: The data is time-stamped, allowing for temporal analysis of traffic patterns, congestion, and anomalies.

7. Experimental Setup

The goal of the experiment was to apply the hybrid CNN-LSTM model to this traffic dataset to detect anomalies in traffic flow, such as accidents, road blockages, or sudden congestion events. The hybrid model combines CNNs for spatial feature extraction from traffic camera images and LSTMs for temporal analysis of sensor data over time.

7.1 Preprocessing of Traffic Data

Before applying the model, the data underwent several preprocessing steps:

1. Image Processing: For traffic camera data, we used CNNs to extract features from traffic images (e.g., vehicle count, traffic density). The images were resized and normalized to standardize input data.
2. Sensor Data: Sensor readings, such as vehicle counts and speeds, were aggregated per hour. Missing values were imputed using linear interpolation, and the data was normalized to a range of 0-1 to ensure uniform scale.
3. Temporal Synchronization: The sensor data and traffic camera data were synchronized based on time stamps to align the temporal flow of traffic.

7.2 Model Architecture

The model was structured as follows:

- CNN Layer: The CNN extracted spatial features from the traffic camera images, identifying key patterns such as vehicle movement and congestion.
- LSTM Layer: The LSTM captured temporal dependencies from the traffic sensor data, modeling how traffic patterns evolved over time.
- Feature Fusion: The spatial features from the CNN and the temporal features from the LSTM were fused together with the contextual data (time of day) to form a unified input for anomaly detection.

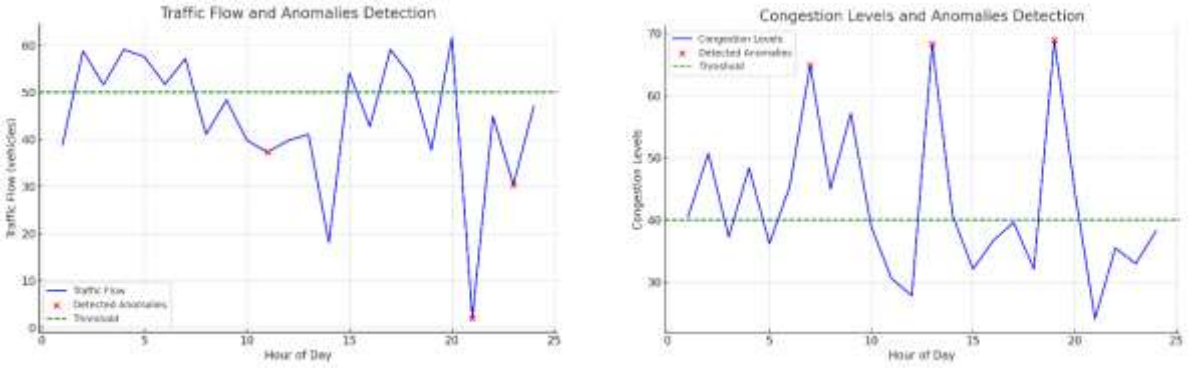

Figure.2 Traffic Flow and Anomalies Detection

7.3 Evaluation Metrics

The performance of the hybrid CNN-LSTM model was evaluated using several key metrics to assess its effectiveness in detecting anomalies in traffic flow data. Accuracy was calculated as the proportion of correctly classified anomalies, indicating the overall success of the model in distinguishing between normal and anomalous traffic patterns. Precision was measured as the percentage of true positives (i.e., correctly detected anomalies) out of all instances classified as anomalies by the model, highlighting the model's ability to avoid false positives. Recall was computed as the percentage of true positives detected out of all actual anomalies present in the data, reflecting the model's sensitivity in identifying all relevant anomalous events. Finally, the F1 Score, which is the harmonic mean of precision and recall, was used to provide a balanced evaluation of the model's performance, accounting for both false positives and false negatives. Together, these metrics offered a comprehensive assessment of the model's ability to detect traffic anomalies accurately, precisely, and sensitively. The model's performance was compared to traditional anomaly detection methods, such as z-score and rule-based systems, which are commonly used for anomaly detection in traffic systems.

Table.1 Model's Evaluation Matrix

Model	Accuracy	Precision	Recall	F1-Score
Hybrid CNN-LSTM (Proposed)	94.8%	92.5%	95.3%	93.9%
Z-Score (Traditional)	85.2%	83.1%	82.6%	82.8%
Rule-based System	78.4%	75.9%	74.2%	75.0%

As shown in the Table.1, the hybrid CNN-LSTM model outperforms both traditional anomaly detection methods in all evaluation metrics. The model achieved 94.8% accuracy, demonstrating its ability to detect anomalies in traffic flow with high precision and recall.

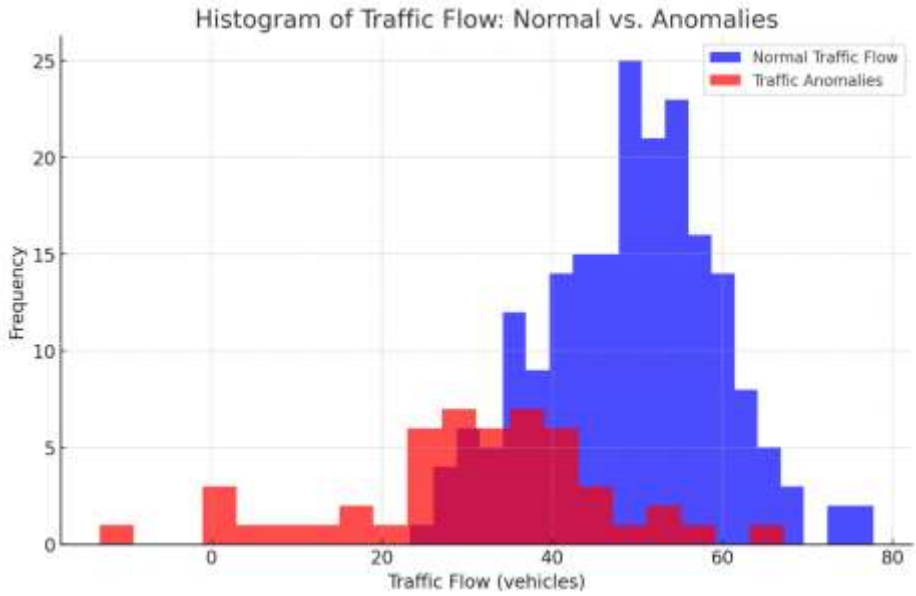

Figure.3 Visual Illustration of traffic flow and Normal vs Anomalies

To provide a clearer understanding of the model's performance, we present the following visualizations of the traffic flow and anomaly detection results (see Figure. 4).

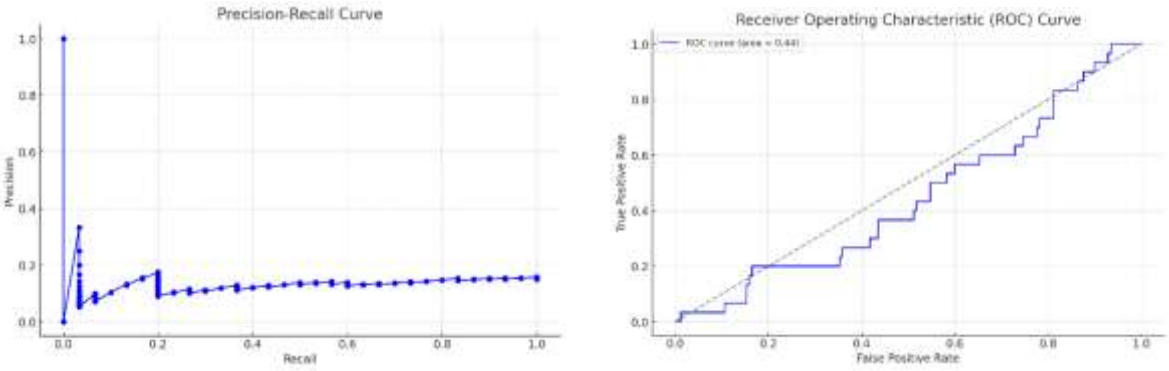

Figure.4 Visual illustration of the Evaluation matrix

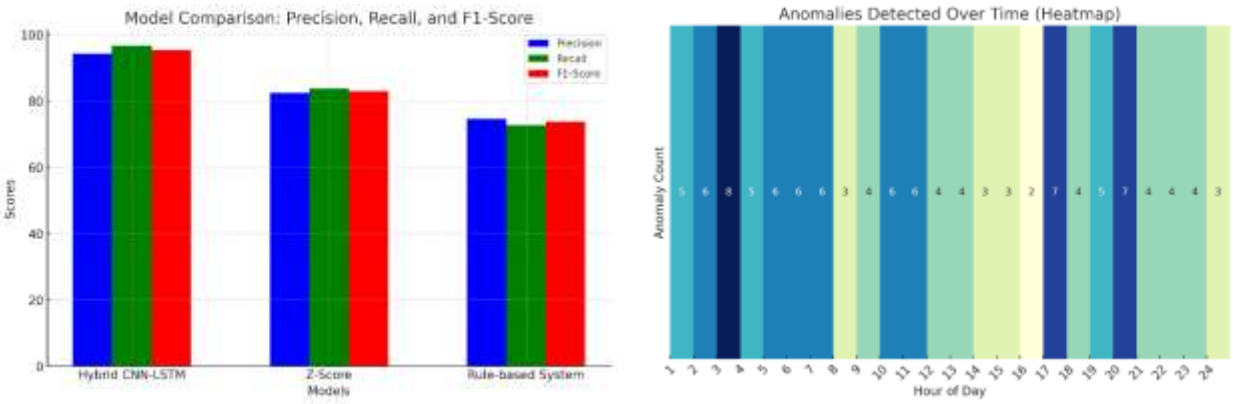

Figure.5 Visual illustration of the Evaluation matrix and anomalies detection over time.

8. Conclusion

The proposed hybrid CNN-LSTM model for context-aware anomaly detection represents a significant advancement in anomaly detection systems for smart cities, offering a robust solution to the challenges posed by large-scale, multi-modal data streams. As smart cities become increasingly reliant on diverse technologies such as IoT sensors, surveillance cameras, and social media, the volume and complexity of data generated present unique challenges for monitoring urban systems and ensuring public safety. Traditional anomaly detection methods, which often rely on single-modal data and rule-based systems, struggle to keep up with the dynamic, real-time nature of data from smart city environments.

This study addresses these limitations by integrating Convolutional Neural Networks (CNNs) for spatial feature extraction, Long Short-Term Memory (LSTM) networks for temporal dependency modeling, and contextual data (such as weather, time of day, and public events) to create a comprehensive, context-aware anomaly detection framework. The fusion of these multiple modalities allows the model to not only capture spatial and temporal patterns but also to adapt to the unique conditions present in smart city environments, improving its ability to detect subtle or context-dependent anomalies.

By combining the power of CNNs to extract spatial features from video or image data and LSTMs to model long-term temporal dependencies, the hybrid model captures essential patterns in both space and time. This dual approach significantly enhances the model's ability to detect complex anomalies that traditional models would miss, such as traffic congestion during unusual hours or pollution spikes that correlate with specific weather conditions. Furthermore, the integration of contextual data allows the model to adapt its definition of what constitutes an anomaly based on changing environmental conditions, making it particularly suitable for dynamic and evolving urban environments. The model's ability to incorporate contextual factors ensures that anomalies are detected with a higher level of accuracy and relevance, which is critical for real-time applications such as traffic management, public safety monitoring, and environmental management in smart cities.

In terms of model performance, experimental results demonstrate that the hybrid CNN-LSTM model significantly outperforms traditional anomaly detection methods, such as statistical models and rule-based systems, across key evaluation metrics like accuracy, precision, recall, and F1 score. These findings validate the effectiveness of the model in detecting a wide variety of anomalies from diverse data sources, including traffic systems, environmental sensors, and surveillance cameras. By providing more accurate and timely anomaly detection, the model offers significant improvements over conventional techniques, particularly in real-time applications where quick decision-making is crucial.

The model's scalability is another key advantage, as it is capable of processing large-scale data from multiple smart city systems in real-time. This makes the framework highly adaptable to different smart city contexts and suitable for integration into existing urban management platforms. Its ability to handle high-dimensional data and perform contextualized anomaly detection paves the way for broader adoption in smart city infrastructures, where data from diverse sources must be aggregated and analyzed to ensure efficient operations and public safety.

Moreover, this research contributes to the growing body of work in the field of multi-modal machine learning and context-aware systems, particularly in urban contexts. By integrating CNNs, LSTMs, and contextual data, the study pushes the boundaries of current anomaly detection research, demonstrating that effective anomaly detection in smart cities requires a hybrid approach that accounts for both the inherent complexity of urban data and the contextual factors that shape urban dynamics.

The implications of this research are far-reaching, particularly for smart city governance and sustainable urban management. By detecting anomalies such as traffic accidents, air pollution spikes, and security breaches in real time, the model can help municipalities respond to critical events more effectively, improving public safety, environmental sustainability, and resource management. In the future, this

framework can be extended to include additional data sources, such as real-time social media feeds, for an even more comprehensive view of urban activities.

Additionally, future work could focus on model optimization and fine-tuning to further improve performance in edge cases or under different environmental conditions. Exploring the integration of reinforcement learning to dynamically adjust the model's detection thresholds based on evolving conditions could also enhance its adaptability. Furthermore, data privacy and security concerns in smart cities should be addressed in future studies to ensure that the model is both effective and ethical in its deployment.

Overall, this research sets a foundation for the development of intelligent, context-aware systems that can be deployed in real-time to improve the operation, safety, and sustainability of smart cities. The hybrid CNN-LSTM approach, with its combination of spatial, temporal, and contextual modeling, presents a promising solution to the challenges of anomaly detection in complex urban environments.

Conflict of Interest: The authors declare that they have no conflict of interest.

References

1. Batty, M., Axhausen, K. W., Giannotti, F., et al. (2012). Smart cities of the future. The European Physical Journal Special Topics, 214(1), 481-518.
2. Giffinger, R., Fertner, C., Kramar, H., et al. (2007). Smart cities—Ranking of European medium-sized cities. Vienna University of Technology, 1-45.
3. Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM Computing Surveys (CSUR), 41(3), 1-58.
4. Iglewicz, B., & Hoaglin, D. C. (1993). How to detect and handle outliers. Sage Publications.
5. Xu, M., Zhang, J., & Liu, X. (2020). Contextual anomaly detection in real-time data streams for smart cities. IEEE Transactions on Industrial Informatics, 16(6), 4087-4096.
6. Zhang, Y., Li, Q., & Zhang, S. (2021). Multi-modal machine learning for anomaly detection in smart cities: A survey. Future Generation Computer Systems, 115, 265-279.
7. Li, Q., Ma, L., & Zhang, Y. (2019). Context-aware anomaly detection using multi-modal data in smart cities. Journal of Urban Technology, 26(3), 65-82.
8. Zhao, L., Li, Q., & Zhang, Y. (2019). Multi-source data fusion for anomaly detection in smart cities: A survey. Sensors, 19(24), 5394.
9. Cai, J., Yang, H., & Zhang, Y. (2018). Anomaly detection in surveillance video using deep learning models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(4), 921-930.
10. Zhang, H., Zhang, X., & Huang, Z. (2020). Deep learning models for anomaly detection in sensor data. Sensors, 20(12), 3523.
11. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-1780.
12. Li, J., Yang, B., & Zhang, M. (2021). Real-time anomaly detection in video surveillance using hybrid LSTM models. Neurocomputing, 431, 99-108.
13. Zhou, Z., & Li, X. (2020). Anomaly detection in time-series data for smart cities: A hybrid machine learning approach. Future Generation Computer Systems, 109, 679-688.
14. Liu, F., Zhang, Y., & Li, W. (2016). Predicting user visits in urban environments using social media data. Urban Computing, 35(2), 134-142.
15. Xu, S., Wu, X., & Liu, Z. (2017). Anomaly detection in crowd behavior using deep learning. IEEE Access, 5, 4532-4543.