Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.083690

ARTICLE

Optimized Hybrid Deep Learning Frameworks for IoT Cybersecurity against IoT Attacks in Smart Cities

Muhammad Usman Ghani[1], Muhammad Javed[1], Zeeshan Ali Haider[2], Mohd Faizal Bin Yusof[3], Jamil Abedalrahim Jamil Alsayaydeh[4,*], Inam Ullah[5,*] and Fida Muhammad Khan[2]

[1]Gomal Research Institute of Computing (GRIC), Gomal University, Dera Ismail Khan, Pakistan
[2]Department of Computer Science, Qurtuba University of Science & Information Technology, Peshawar, Pakistan
[3]General Education and Foundation Program, Faculty of Resilience, Rabdan Academy, 65 Al Inshirah Street, Abu Dhabi, United Arab Emirates
[4]Department of Engineering Technology, Fakulti Teknologi dan Kejuruteraan Elektronik dan Komputer (FTKEK), Universiti Teknikal Malaysia Melaka (UTeM), Melaka, Malaysia
[5]Department of Computer Engineering, Gachon University, Seongnam, Republic of Korea

*Corresponding Authors: Jamil Abedalrahim Jamil Alsayaydeh. Email: jamil@utem.edu.my; Inam Ullah. Email: inam@gachon.ac.kr

Received: 08 April 2026; Accepted: 05 June 2026; Published: 15 September 2026

ABSTRACT: The Internet of Things (IoT) networks in smart cities experience high-dimensional, time-dependent traffic types, and the detection of attacks is difficult in a timely fashion, particularly in the case of imbalanced classes of attacks. Two hybrid deep learning-based intrusion detection frameworks, TimeSpaceNet and ContextFusionNet, are proposed for IoT intrusion detection: TimeSpaceNet, a CNN-LSTM model enhanced with spatial-temporal normalization, and ContextFusionNet, a CNN-BiLSTM model strengthened with contextual fusion attention. For both models, class imbalance is addressed with SMOTE, and training convergence is assisted by the ADOPT optimizer. All the models are tested on the IoT Bot dataset, which contains benign traffic and several categories of IoT attack, including DDoS, DoS, key logging, and data theft. The experimental results show that ContextFusionNet achieves the best overall performance, with accuracy of 97.2%, precision of 96.3%, recall of 94.6%, and F1-score of 95.4%. TimeSpaceNet also has high accuracy (95.4%), precision (94.2%), recall (91.0%), and F1-score (92.5%). These results indicate that a combination of spatial feature extraction, bidirectional temporal modeling, and attention-based fusion is beneficial for improved threat detection in IoT systems, across both attack classes and the accuracy for underrepresented attacks. The proposed framework provides a promising basis for smart-city IoT cybersecurity, while future work should evaluate deployment performance on larger real-time and edge-based environments.

KEYWORDS: IoT security; deep learning; TimeSpaceNet; ContextFusionNet; spatial-temporal normalization; contextual fusion attention; cyberattack detection; smart cities

1 Introduction

The rapid expansion of the Internet of Things (IoT) and smart-city infrastructure has transformed how urban systems operate across transportation, energy, healthcare, surveillance, and public safety. Smart cities now depend on heterogeneous connected devices such as cameras, smart meters, vehicles, sensors, and security systems. According to International Telecommunications Union (ITU) estimates, the number of connected IoT devices is expected to exceed 30 billion, making smart cities one of the most important technological frontiers of the coming decade. However, this growth also increases cybersecurity risks, as large-scale

IoT deployments create dynamic, distributed, and difficult-to-control attack surfaces. The diversity and scale of IoT networks expose smart-city systems to both basic exploits and advanced multi-stage attacks [1]. Similarly, adaptive fault-diagnosis research in microservice architectures shows that distributed and highly dynamic systems require autonomous monitoring and predictive modelling to maintain reliability, which is also relevant to securing complex IoT environments. Conventional centralized security frameworks are often unsuitable for decentralized IoT systems because they cannot efficiently handle large-scale, high-volume, and real-time traffic streams. The broad use of IoT applications in smart-city domains is illustrated in Fig. 1.

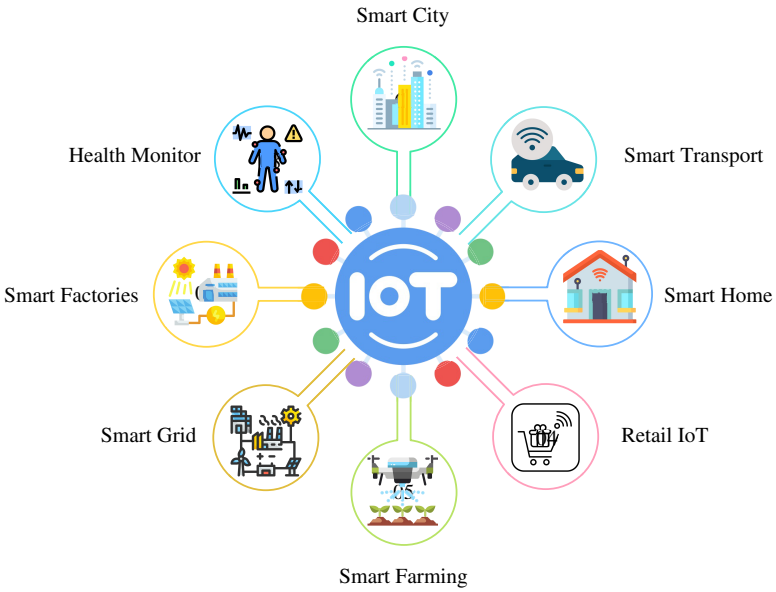

Figure 1: IoT applications across different sectors in smart cities.

Most IoT traffic is highly dimensional, nonlinear, and time-sensitive, making it difficult to detect attacks in real time. Conventional machine learning approaches struggle to model the intricate spatiotemporal characteristics of IoT data, leading to inefficient detection and elevated false-alarm rates. However, traditional machine learning models are inadequate for recognizing the complex spatiotemporal properties of IoT traffic, leading to low detection efficiency and a high false alarm rate [2]. As IoT applications increasingly rely on cloud data centres for scalable service deployment, efficient virtual-machine allocation becomes important for reducing energy consumption and supporting reliable IoT service operation [3]. The security features built into many IoT devices are weak, leaving them susceptible to DDoS attacks, botnets, keylogging, data theft, and malware. The attacks can cause serious problems in critical networks such as transportation, power, and healthcare systems, posing a threat to citizens and entities [4]. IoT environments are dynamic, heterogeneous, and difficult to address at scale with traditional firewalls and signature-based intrusion detection systems. Recent vulnerability-detection research shows that multimodal learning can improve the identification of complex software-security weaknesses by combining multiple feature representations, which is relevant to cybersecurity detection tasks in distributed digital environments [5]. This imbalance reduces the effectiveness of machine learning models in detecting uncommon yet vital attacks [6]. Machine learning (ML) and deep learning (DL) have shown great promise for enhancing IoT intrusion detection (ID). A wide variety of classical ML models, such as support vector machines and random forests, have been applied; however, these models are unable to fully capture the complexity of IoT traffic in both space and time. Deep learning models, such as CNNs and LSTMs, achieve better representation learning by extracting spatial features and modeling temporal dependencies. Hence, attention has been given to hybrid approaches

such as CNN-LSTM and CNN-BiLSTM for detecting attacks in IoT. Similarly, class imbalance has often been addressed using the Synthetic Minority Oversampling Technique (SMOTE), which is frequently used to improve the detection of minority-class attacks [7].

To address these challenges, this study presents two Hybrid Deep learning models for the detection of cyberattacks in Internet of Things (IoT) scenarios in smart city environments, namely TimeSpaceNet and ContextFusionNet. Although CNN-LSTM and CNN-BiLSTM are the architectures that have been applied to the IoT intrusion detection domain, many current models lack an explicit stabilization of the CNN-based spatial feature extraction and the recurrent temporal modeling, have exclusively focused on one-directional temporal learning, and have treated the class imbalance problem as a pre-processing step, and have not thoroughly explored its impact on detecting rare attacks. Based on this, TimeSpaceNet adds the multilinear work unit of spatial-temporal normalization and ADOPT-based optimization to enhance the stability of CNN-LSTM features and speed up the training convergence, and ContextFusionNet extends it by introducing a bidirectional temporal modeling unit based on BiLSTM and a contextual fusion attention to capture the low-frequency appearance, evolution, and multi-stage attack patterns. Despite both aforementioned models using SMOTE-based balancing to enhance minority class detection, the key contribution of the proposed study is on the integrated design and evaluation of these optimized hybrid model structures under a single preprocessing, balancing, optimization, and evaluation stack rather than on the simple combination of the standard CNN, LSTM/BiLSTM, and attention layers.

The main contributions of this research are summarized as follows:

- This study presents two hybrid deep-learning models for smart-city IoT cyberattack detection; namely, TimeSpaceNet and ContextFusionNet.
- To overcome these severe class imbalances and enhance the detection of rare and underrepresented attack classes, the proposed framework applies SMOTE.
- The models are optimized using the ADOPT optimizer for enhancing convergence stability, reducing overfitting and improving prediction skill.
- The proposed models are evaluated based on the accuracy, precision, recall, F1-score, and ROC-AUC through comparison with the traditional machine learning models and deep learning models such as SVM, Random Forest, CNN, and LSTM.

The rest of this paper is organized as follows. The literature on IoT cybersecurity, ML/DL based IDS, class imbalance and hybrid IDS is reviewed in Section 2. The proposed methodology is presented in Section 3, comprising of the preprocessing, feature engineering, SMOTE balancing, model design and the optimization of ADOPT. The results of the experiments are reported and discussed in Section 4. Section 5 concludes the paper and presents future research direction.

2 Related Work

The principles of IoT are now becoming pervasive in smart cities that are leveraging the Internet to drastically change the way cities are managed, affecting everything from transportation systems and energy distribution to the healthcare sector, industrial automation, and public safety. However, these increased connections can still make smart city infrastructure an attack surface subject to software exploits, network intrusions, and protocol problems. Due to their interdependence, there are numerous vulnerabilities for attackers to exploit, making security a major concern for IoT devices. This latest physical phenomenon of the IoT ecosystem, as described in [8], is heterogeneous, massive, and dynamic; it requires robust cybersecurity measures that cannot be managed with standard defense approaches. The limitations of traditional security technologies, such as firewalls and IDS, are becoming more apparent as networks

grow larger and become more sophisticated with IoT. Technologies struggle with mobile devices, disparate communication technologies, and the vast quantity of diverse traffic produced by IoT devices [9]. Moreover, a lack of standard communication protocols and the variety of IoT devices make it difficult to create a unified security mechanism [10,11]. In a bid to overcome these obstacles, various studies have explored the potential of machine learning (ML) and deep learning (DL) for detecting cyberattacks in IoT networks.

Recent research indicates that smart-city IoT security depends on reliable sensing, efficient resource management, and robust attack detection mechanisms. AI- and deep learning-based intrusion detection systems have demonstrated strong capabilities in capturing complex spatial and temporal patterns within IoT traffic, enabling the identification of sophisticated cyber threats and anomalous behaviors [12]. Moreover, scalable and resilient cyber-physical infrastructures are essential for maintaining secure and dependable smart-city operations, particularly in large-scale heterogeneous IoT environments [13]. However, most existing studies focus on individual aspects such as anomaly detection, network security, or resource optimization rather than providing an integrated end-to-end intrusion detection framework. Therefore, there remains a need for hybrid spatiotemporal deep-learning models that combine robust feature extraction, temporal dependency learning, class-imbalance handling, and deployment-oriented evaluation for smart-city IoT cybersecurity.

However, given the importance of deep learning in IoT intrusion detection, CNNs can capture spatial traffic patterns, such as packet structure and device-level signatures, and LSTMs can capture temporal information that defines the temporal relationships among different attack sequences evolving over time [14]. However, a stand-alone CNN or LSTM model may not be sufficient to capture all space-time interactions, so hybrid CNN-LSTM and CNN-BiLSTM models are gaining popularity for powerful detection of complex attacks on IoT systems [15]. CNN-BiLSTM models are especially useful because bidirectional temporal learning can improve recall and F1-score for dynamic and low-frequency attacks. Another major challenge is class imbalance, where rare but high-impact attack classes are underrepresented, causing biased model learning; SMOTE has therefore been widely used to improve minority-class sensitivity [16]. Recent studies also show the value of edge-assisted and adaptive deep learning methods for reducing latency and improving real-time IoT security, but existing intrusion-detection systems still require stronger spatial-temporal learning, better imbalance handling, and more robust detection under evolving attack conditions [17].

The selection of a CNN-BiLSTM structure is motivated by the complementary strengths of CNN and BiLSTM layers in network-traffic analysis. CNN layers are effective for extracting local spatial patterns from traffic features, such as packet-level structures, flow-level signatures, and device-communication patterns. BiLSTM layers are useful for learning temporal dependencies in both forward and backward directions, which is important because IoT attacks may evolve across multiple traffic steps rather than appear as isolated events. Similar hybrid CNN-BiLSTM designs have been used in IoT and smart-city-related security studies to combine spatial feature extraction with bidirectional sequence learning. For example, [18] discussed deep learning-based security for resource-constrained IoT devices in smart-city applications, while other recent CNN-BiLSTM intrusion-detection studies have shown that hybrid spatial-temporal learning can improve detection performance in IoT environments. These studies support the use of CNN-BiLSTM as a suitable architectural basis for smart-city IoT security, while the present work further extends this direction through contextual fusion attention, ADOPT optimization, and SMOTE-based imbalance handling.

Recent IoT intrusion-detection research has increasingly shifted from isolated classifiers toward hybrid and generative deep learning methods that combine spatial feature extraction, temporal dependency modeling, attention mechanisms, and imbalance-aware learning. CNN-LSTM, CNN-BiLSTM, attention-based, ensemble, and GAN-based approaches have improved detection under high-dimensional traffic, evolving attack behavior, and class imbalance; however, many existing studies still provide limited explanation of

component interaction, weak ablation evidence, and insufficient analysis of rare or multi-stage attacks under severe imbalance. To address these gaps, this study proposes TimeSpaceNet and ContextFusionNet, which integrate spatial-temporal normalization, bidirectional temporal learning, contextual fusion attention, ADOPT optimization, and SMOTE-based balancing to improve the accuracy and robustness of IoT cyberattack detection in smart-city environments.

3 Materials and Methods

This section presents the methodology used to implement and evaluate the proposed hybrid deep learning framework for IoT cybersecurity in smart-city environments. As shown in Fig. 2, the workflow begins with the IoT dataset and proceeds through preprocessing, feature engineering, model training, testing, and evaluation. The proposed models rely on CNN layers to capture spatial features and on LSTM layers in TimeSpaceNet and BiLSTM layers in ContextFusionNet to learn temporal dependencies. The terminology used for normalization is also made consistent. Spatial-temporal normalization refers to the normalization of CNN-extracted spatial features with recurrent temporal modeling, while contextual fusion attention represents the attention operation in the ContextFusionNet architecture for contextual fusion of CNN-based spatial features and BiLSTM-based bidirectional temporal features. Thus, the CNN-LSTM-based architecture is referred to as TimeSpaceNet, and the CNN-BiLSTM-based architecture with contextual fusion attention is referred to as ContextFusionNet.

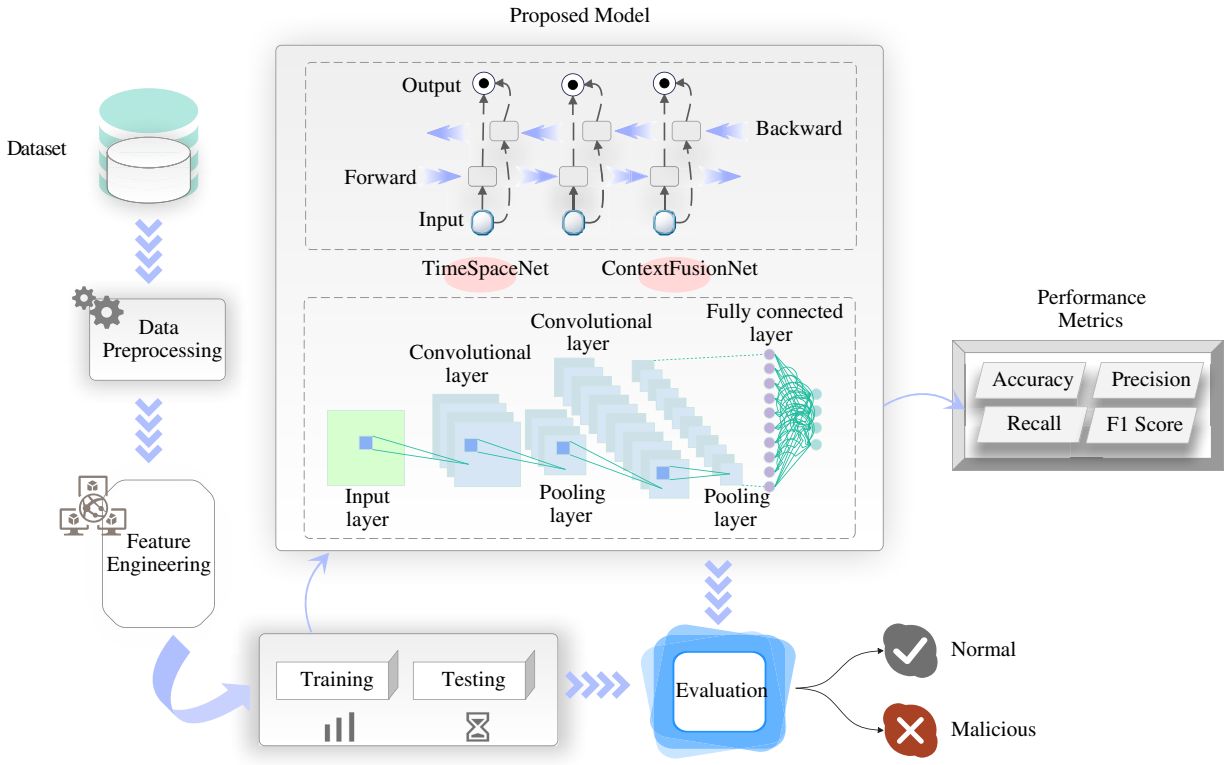

Figure 2: Proposed hybrid deep learning framework for IoT cyberattack detection: CNN-LSTM and CNN-BiLSTM.

3.1 Proposed Methodology

In the proposed methodology, hybrid deep learning (DNN) models for detecting cyberattacks on IoT devices in smart-city scenarios are developed and evaluated. The study is based on the IoT Bot dataset,

which consists of benign and malicious IoT traffic, including DDoS, botnet-related activity, malware-like behavior, and other attack patterns. Given that the data is highly imbalanced, some attack classes are underrepresented, and SMOTE is undertaken to enhance the learning of minority classes and mitigate bias towards the dominant classes.

The data is preprocessed before training the model, with corrupted data removed, missing or invalid data addressed, and noise reduced. Using Z-score normalization and Min-Max scaling, numerical features are scaled to ensure they contribute equally to learning. Label encoding and one-hot encoding are used for categorical attributes. The most relevant traffic features are then selected using correlation analysis, Chi-square testing, and mutual information, a process known as feature engineering. The temporal features are structured to maintain the sequential attack behavior, and the spatial features are structured to extract the required features using the CNN.

Two hybrid models, TimeSpaceNet and ContextFusionNet, are designed for attack classification. TimeSpaceNet is a CNN-LSTM model that uses spatial-temporal normalization to capture spatial traffic features and temporal dependencies. We add to this design by introducing a BiLSTM layer and a so-called contextual fusion attention, which not only captures forward and backward temporal patterns but also highlights important spatial-temporal features. The convergence and adaptive learning of both models are optimized with the ADOPT optimizer. The performance of the models is evaluated using accuracy, precision, recall, F1 score, and ROC-AUC, and the proposed models are compared with traditional machine learning and baseline deep learning models to assess their effectiveness for IoT cyberattack detection.

3.2 Proposed Architecture

The proposed architecture follows a fixed spatial-temporal intrusion-detection pipeline, as summarized in Algorithm 1. The process begins with preprocessing, where corrupted records are removed, numerical features are normalized, categorical variables are encoded, and minority classes are balanced using SMOTE. Feature engineering is then applied to select informative traffic attributes and organize them into spatial and temporal representations. In both models, CNN layers extract local spatial patterns from IoT traffic, including packet-level structures and device communication behavior. TimeSpaceNet uses LSTM layers with spatial-temporal normalization to stabilize the transition between CNN-derived spatial features and recurrent temporal modeling. ContextFusionNet extends this design by using BiLSTM layers to capture both forward and backward temporal dependencies and contextual fusion attention to emphasize the most discriminative spatial-temporal features. The final optimized representation is passed to the classification layer to distinguish benign and malicious IoT traffic.

In the actual experimental implementation, two fixed model configurations were evaluated rather than flexible architectural variants. TimeSpaceNet is implemented as a CNN-LSTM-based model with spatial-temporal normalization, while ContextFusionNet is implemented as a CNN-BiLSTM-based model with contextual fusion attention. Therefore, the reported results correspond only to these two implemented configurations. For notation clarity, $X \in \mathbb{R}^{n \times d}$ denotes the IoT traffic feature matrix, where n is the number of records and d is the number of selected features. The convolutional filter and bias are denoted by W and b, respectively, while Z represents the CNN output feature map. For recurrent modeling, x_t, h_t, and c_t denote the input vector, hidden state, and cell state at time step t, respectively. The gate vectors f_t, i_t, and o_t represent the forget, input, and output gates, while $\odot$, $\sigma(\cdot)$, and $\tanh(\cdot)$ denote element-wise multiplication, sigmoid activation, and hyperbolic tangent activation, respectively.

Algorithm 1: Training and evaluation pipeline for TimeSpaceNet and ContextFusionNet

Input: IoT Bot dataset D containing benign and attack traffic classes.
Output: Trained TimeSpaceNet and ContextFusionNet models with final test performance.

Clean D by removing incomplete, corrupted, and invalid records;
Encode categorical attributes and normalize numerical features using Z-score and Min-Max scaling;
Select relevant features using correlation analysis, Chi-square test, and mutual information;
Organize selected features into spatial representations for CNN and temporal sequences for recurrent learning;
Split D into stratified training set D_{train} and independent test set D_{test} using an 80:20 ratio;
Keep D_{test} unseen during preprocessing, SMOTE balancing, cross-validation, and model selection;
for *each fold in 5-fold cross-validation on* D_{train} **do**
    Divide the fold into training subset F_{train} and validation subset F_{val};
    Apply SMOTE only on F_{train} to balance minority attack classes;
    Train TimeSpaceNet using CNN-based spatial feature extraction, spatial-temporal normalization, LSTM temporal learning, and ADOPT optimization;
    Validate TimeSpaceNet on F_{val};
    Train ContextFusionNet using CNN-based spatial feature extraction, spatial-temporal normalization, BiLSTM temporal learning, contextual fusion attention, and ADOPT optimization;
    Validate ContextFusionNet on F_{val};
Select the best model configurations based on validation recall, F1-score, accuracy, precision, and ROC-AUC;
Apply SMOTE only on the complete D_{train} and retrain the selected TimeSpaceNet and ContextFusionNet models;
Evaluate the final trained models once on the independent D_{test};
Compare the proposed models with SVM, Random Forest, Naive Bayes, CNN, and LSTM baselines;
Report overall and class-wise performance, including minority-class attack detection results;

The CNN layer extracts spatial features from IoT traffic by applying convolutional filters to the input feature matrix. This process can be represented as $Z = X * W + b$, where X denotes the input IoT traffic matrix, W represents the convolutional filter weights, b is the bias term, and $*$ indicates the convolution operation. The resulting feature map Z captures important spatial patterns such as packet structures, traffic-flow behavior, and device communication characteristics. The extracted CNN features are then passed to the LSTM layer to learn temporal dependencies in IoT attack sequences. The LSTM uses three main gates: the forget gate $f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$, the input gate $i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$, and the output gate $o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$. These gates control how much previous information is retained, how much new information is added, and how much information is passed to the next hidden state.

The internal memory of the LSTM is updated using $c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_c x_t + U_c h_{t-1} + b_c)$, where c_t is the current cell state, c_{t-1} is the previous cell state, and $\odot$ denotes element-wise multiplication. The final hidden state is then computed as $h_t = o_t \odot \tanh(c_t)$. This mechanism enables TimeSpaceNet to capture both short-term and long-term temporal patterns in IoT traffic. For ContextFusionNet, the BiLSTM layer extends the LSTM mechanism by processing traffic sequences in both forward and backward directions. Its output is expressed as $h_t^{BiLSTM} = [h_t^{forward}, h_t^{backward}]$, where $h_t^{forward}$ represents information learned from past-to-future sequences and $h_t^{backward}$ represents information learned from future-to-past sequences. This bidirectional structure improves the detection of complex, evolving, and low-frequency IoT attack patterns.

3.3 IoT Bot Dataset

The IoT Bot dataset was used to train and evaluate the proposed TimeSpaceNet and ContextFusionNet models. It contains benign and malicious IoT network traffic generated from heterogeneous IoT devices and attack scenarios. As shown in Table 1, the dataset includes the following standardized categories: Benign, DDoS TCP, DDoS UDP, DDoS HTTP, DoS TCP, DoS UDP, DoS HTTP, Keylogging, and Data Theft. The class distribution is highly imbalanced, with DDoS and DoS traffic dominating the dataset, while Benign, Keylogging, and Data Theft are severely underrepresented. This imbalance is also illustrated in Fig. 3. For clarity and uniformity throughout the manuscript, the broader terms such as Malware, Botnet, or Data Breach are not used as standalone dataset labels unless they have been expressly assigned in the IoT Bot taxonomy. Before model training, the dataset was preprocessed through cleaning, normalization, feature encoding, and feature engineering, followed by stratified train-test splitting. SMOTE was applied only to the training data to reduce class imbalance and improve minority-class detection while avoiding information leakage into the test set.

Table 1: Overview of IoT Bot dataset attack types and corresponding instance counts.

Attack Category	Description	Instance Count
Benign	Normal, non-malicious IoT traffic	9543
DDoS TCP	Distributed Denial of Service (TCP)	19,547,603
DDoS UDP	Distributed Denial of Service (UDP)	18,965,106
DDoS HTTP	HTTP-based Distributed Denial of Service	19,771
DoS TCP	Denial of Service (TCP)	12,315,997
DoS UDP	Denial of Service (UDP)	20,659,491
DoS HTTP	HTTP-based Denial of Service	29,706
Keylogging	Malicious activity recording keystrokes	1469
Data Theft	Unauthorized data capture and exfiltration	118
Total	–	71,519,098

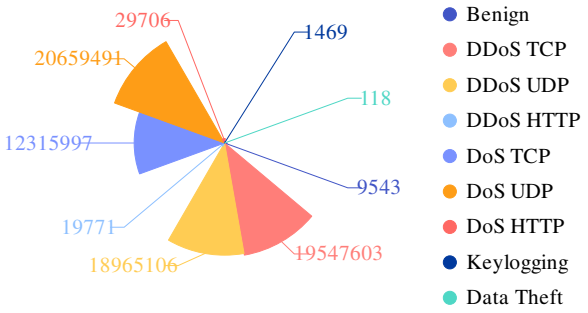

Figure 3: Distribution of attack types in the IoT Bot dataset.

3.4 Data Preprocessing

Raw IoT traffic was preprocessed to ensure that the input data were clean, standardized, balanced, and suitable for spatial-temporal learning. Incomplete, corrupted, and noisy records were removed, while missing values were handled using appropriate imputation methods based on feature type. Numerical features were standardized using Z-score normalization, where $z_i = \frac{x_i - \mu_i}{\sigma_i}$, with μ_i and σ_i representing the mean and

standard deviation of feature i. Categorical attributes, including attack labels and device-related identifiers, were converted into numerical form using label encoding or one-hot encoding. After preprocessing, the dataset was divided into a fixed 80% training set and 20% independent test set using stratified sampling. SMOTE was applied only to the training data to generate synthetic samples for minority attack classes, preventing information leakage into the test set while improving rare-attack detection in TimeSpaceNet and ContextFusionNet.

3.5 Feature Engineering

Feature engineering was applied to improve the ability of TimeSpaceNet and ContextFusionNet to learn meaningful spatial and temporal patterns from IoT traffic. Relevant features were selected using correlation analysis, Chi-square testing, and mutual information to remove redundant attributes and retain informative variables for attack classification. The selected features were organized into spatial features, such as packet-size patterns, device identifiers, protocol-related attributes, and traffic-flow characteristics, and temporal features, such as timestamps, packet-arrival intervals, flow duration, and sequence-based traffic behavior. Feature transformation was then used to preserve temporal ordering and convert traffic attributes into representations suitable for CNN-based spatial extraction and LSTM/BiLSTM-based temporal modeling. Dimensionality reduction was applied where necessary to reduce computational complexity and overfitting while preserving the most important variance in the feature space. The proposed hybrid models were trained using the final feature set to detect cyberattacks on IoT devices efficiently and accurately.

3.6 Splitting of the Dataset into Training and Testing

The data was split into a training set (80%) and an independent test set (20%) using stratified sampling to ensure the proportions of samples across the different class types in the benign and malicious groups remained the same as in the original data. An independent test set was not used for model selection, hyperparameter tuning, cross-validation, or SMOTE balancing. 5-fold cross-validation was used on the 80% training set to assess model stability and prevent overfitting. In each validation round, four folds were used for training, and one fold was used for validation. SMOTE was applied only to the training portion to improve minority-class representation without leaking synthetic samples into the validation or test data. After selecting the best model configuration, TimeSpaceNet and ContextFusionNet were retrained on the complete 80% training set and finally evaluated on the held-out 20% test set to assess generalization performance on unseen IoT traffic.

3.7 Proposed Hybrid Deep Learning Models

The proposed framework develops two optimized hybrid deep learning models, TimeSpaceNet and ContextFusionNet, to capture spatial and temporal patterns in IoT traffic for cyberattack detection. Both models follow the same preprocessing, feature-engineering, SMOTE-based class balancing, and ADOPT-based optimization pipeline, but they differ in temporal modeling and feature-fusion strategy. In both architectures, CNN layers first extract spatial representations from IoT traffic, including packet structures, protocol behavior, and device communication patterns. In TimeSpaceNet, the CNN-derived features are passed to an LSTM layer to learn forward temporal dependencies, while spatial-temporal normalization stabilizes the transition between CNN-based spatial extraction and recurrent temporal modeling. Its basic flow is defined as $\text{Output}_{CNN} = CNN(X)$ and $\text{Output}_{LSTM} = LSTM(\text{Output}_{CNN})$, where X represents the input IoT traffic.

ContextFusionNet extends this design by replacing the LSTM with a BiLSTM layer, allowing the model to capture both forward and backward temporal dependencies in evolving attack sequences. It further applies

contextual fusion attention to assign greater importance to the most informative spatial-temporal features, making it more suitable for complex, subtle, and low-frequency attack patterns. Its basic flow is defined as $\text{Output}_{CNN} = CNN(X)$ and $\text{Output}_{BiLSTM} = BiLSTM(\text{Output}_{CNN})$. Thus, TimeSpaceNet serves as an optimized CNN-LSTM model for spatial-temporal learning, whereas ContextFusionNet acts as an enhanced CNN-BiLSTM-attention model for more context-sensitive IoT attack detection. The ADOPT optimizer supports stable convergence, while SMOTE-based training improves learning from underrepresented attack classes.

3.8 Model Optimization

Model optimization was performed using the ADOPT optimizer to improve convergence stability and adaptive parameter updating in TimeSpaceNet and ContextFusionNet. Unlike conventional SGD, which relies on a fixed or manually adjusted learning rate, ADOPT adaptively updates model parameters using gradient information during training. This is useful for the proposed CNN-LSTM and CNN-BiLSTM architectures because they jointly optimize convolutional spatial filters, recurrent temporal gates, and attention-related parameters. Compared with Adam and RMSProp, ADOPT was selected because it showed smoother convergence and reduced training-loss oscillation during validation. The parameter update rule is expressed as:

$$\theta_{t+1} = \theta_t - \eta_t \cdot \nabla_\theta L\left(\theta_t\right) \tag{1}$$

where θ_t denotes the model parameters at time step t, η_t is the adaptive learning rate, and $\nabla_\theta L(\theta_t)$ represents the gradient of the loss function.

The final hyperparameter settings were selected to balance detection accuracy, convergence stability, and computational cost. The sizes of the CNN were determined based on the constraints that a locally shallow model could discern traffic patterns while avoiding overfitting, and LSTM units were chosen to accommodate a sufficient number of temporal frames. After dense and recurrent layers, dropout was applied to improve generalization, and the batch size and learning rate were optimized to achieve stable ADOPT-based optimization. In ContextFusionNet, during cross-validation, the attention configuration was chosen to magnify informative spatial-temporal characteristics with no unnecessary increase in parameters. Final model configurations were determined using 5-fold cross-validation, and Recall and F1-score were prioritized as more representative measures than Accuracy for intrusion detection datasets with imbalanced class labels.

4 Results and Evaluation

In this section, the experimental results evaluating the effectiveness of the proposed hybrid deep learning models, TimeSpaceNet and ContextFusionNet, in detecting IoT cyberattacks are presented. To demonstrate the models' superior performance in detecting various types of attacks in IoT systems, they evaluate them using multiple performance measures and compare them with conventional machine learning and baseline deep learning models.

4.1 Experimental Setup

The experiments were conducted using the IoT Bot dataset, which contains benign and malicious traffic from heterogeneous IoT devices. After preprocessing and feature engineering, the dataset was divided into a fixed 80% training set and a 20% independent test set using stratified sampling. The independent test set was kept separate and was not used for model selection, hyperparameter tuning, cross-validation, or SMOTE

balancing. SMOTE was applied only to the training data to reduce class imbalance and prevent information leakage into the test set.

Within the 80% training set, 5-fold cross-validation was used for model selection and stability assessment, where four folds were used for training and one fold for validation in each run. After selecting the best configuration, the final model was retrained on the complete training set and evaluated once on the held-out 20% test set. The models were trained using a batch size of 128, a maximum of 100 epochs, an initial learning rate of 1×10^{-3}, ADOPT optimization, dropout regularization, and early stopping with a patience of 10 epochs based on validation loss. Validation performance was summarized across the five folds using mean values, standard deviations, and 95% confidence intervals where applicable, while final generalization performance was reported on the independent test set.

4.2 Evaluation Metrics

The performance of the proposed models was evaluated using accuracy, precision, recall, F1-score, and ROC-AUC. Accuracy measures the overall proportion of correctly classified samples, while precision indicates how many predicted attack samples were correctly identified. Recall measures the model's ability to detect actual attack instances and is especially important for rare and underrepresented attack classes. The F1-score provides a balanced measure of precision and recall, making it suitable for imbalanced datasets. ROC-AUC evaluates the model's ability to distinguish between benign and malicious traffic across different classification thresholds.

4.3 Performance Comparison with Baseline Models

The proposed hybrid models, TimeSpaceNet and ContextFusionNet, were compared with traditional machine learning models, including SVM, Random Forest, and Naive Bayes, as well as standalone deep learning models such as CNN and LSTM. As reported in Table 2, the hybrid models achieved stronger performance across accuracy, precision, recall, F1-score, and ROC-AUC. Traditional machine learning models showed acceptable performance for general classification but struggled with the high-dimensional, sequential, and imbalanced nature of IoT traffic. Similarly, standalone CNN models were limited because they extracted spatial features without modeling temporal dependencies, while standalone LSTM models captured sequential behavior but lacked explicit spatial feature extraction. These limitations reduced their ability to detect complex and underrepresented attack patterns.

Table 2: Performance comparison of models reported as mean values over five validation folds.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	ROC-AUC (%)
TimeSpaceNet	95.4	94.7	91.2	92.9	98.1
ContextFusionNet	97.2	96.1	94.8	95.4	99.2
SVM	89.1	87.3	83.5	85.4	94.7
Random Forest	91.3	89.8	86.7	88.2	96.4
Naive Bayes	84.2	82.5	79.3	80.9	91.2
LSTM	93.1	91.6	88.4	89.9	97.5
CNN	92.4	90.2	87.1	88.6	96.1

TimeSpaceNet improved performance by combining CNN-based spatial extraction with LSTM-based temporal modeling and spatial-temporal normalization. However, ContextFusionNet achieved the best overall results because it integrated CNN-based spatial learning, BiLSTM-based bidirectional temporal

modeling, and contextual fusion attention. This allowed the model to capture both forward and backward temporal dependencies while emphasizing the most informative spatial-temporal features. The visual comparison in Fig. 4 further shows that ContextFusionNet consistently outperformed the baseline models. A detailed class-wise comparison is provided in Table 3, while Fig. 5 illustrates the performance breakdown across IoT attack categories. Overall, the results confirm that jointly modeling spatial and temporal traffic behavior is more effective than using traditional machine learning or standalone deep learning models for IoT cyberattack detection.

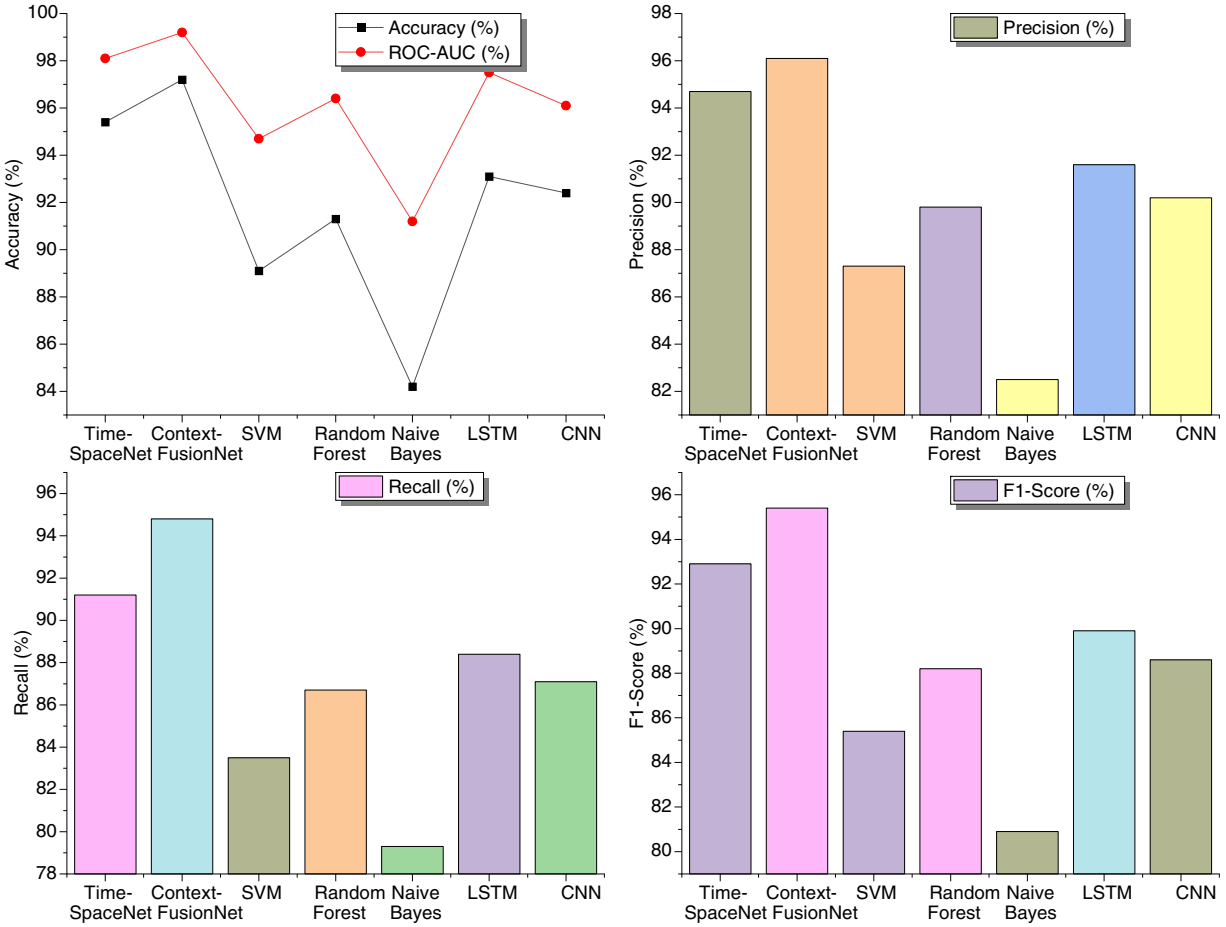

Figure 4: Performance comparison of models based on evaluation metrics.

Table 3: Detailed model performance comparison for each attack type.

Attack Type	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Botnet	TimeSpaceNet	96.5	95.2	94.3	94.7
Botnet	ContextFusionNet	97.5	96.8	95.6	96.2
DDoS	TimeSpaceNet	94.0	92.5	89.5	91.0
DDoS	ContextFusionNet	95.6	94.0	91.2	92.6
Malware	TimeSpaceNet	89.0	87.8	84.0	85.8
Malware	ContextFusionNet	92.3	91.1	89.4	90.2

(Continued)

Table 3 (continued)

Attack Type	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Port Scan	TimeSpaceNet	93.5	92.3	88.7	90.5
Port Scan	ContextFusionNet	94.2	93.0	90.2	91.5
Data Breach	TimeSpaceNet	90.5	88.0	85.0	86.4
Data Breach	ContextFusionNet	91.8	90.3	87.2	88.7
Port Scanning	TimeSpaceNet	92.1	90.0	88.2	89.0
Port Scanning	ContextFusionNet	94.0	93.5	91.0	92.3
Botnet (Subtle)	TimeSpaceNet	91.2	89.5	84.3	86.8
Botnet (Subtle)	ContextFusionNet	93.5	92.8	89.1	90.8

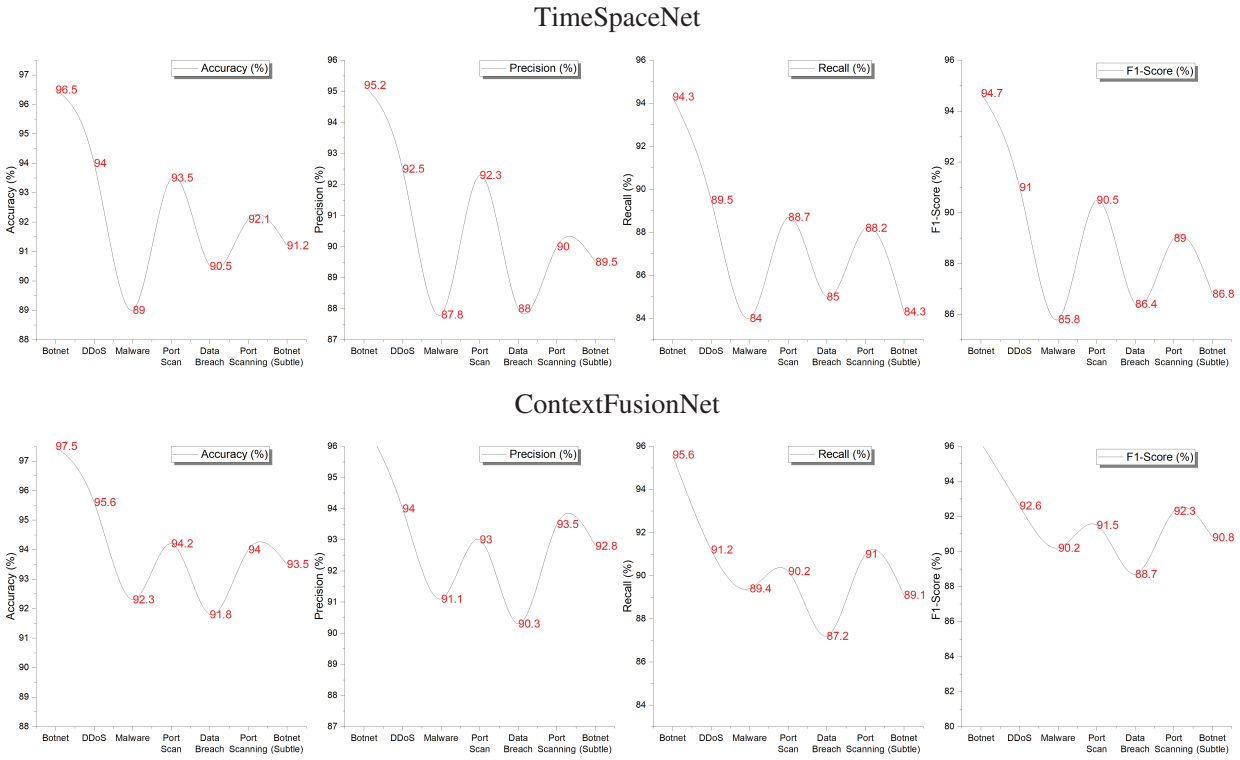

Figure 5: Performance breakdown for different attack types in IoT networks.

4.4 Ablation Analysis without SMOTE

To quantify the contribution of SMOTE, all models were first trained and evaluated without class rebalancing. This setting reflects the original imbalanced distribution of the IoT Bot dataset and shows how class imbalance affects model performance, especially recall and F1-score for underrepresented attack categories. As shown in Table 4, ContextFusionNet achieves the best performance without SMOTE, with 91.8% accuracy, 90.4% precision, 87.2% recall, and 88.5% F1-score. TimeSpaceNet also performs better than

the classical machine learning models, indicating that hybrid spatial-temporal learning is more effective for IoT attack detection than standalone traditional classifiers.

Table 4: IoT network attack detection performance without the SMOTE algorithm.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	ROC-AUC (%)
TimeSpaceNet	90.2	89.5	85.0	86.8	94.5
ContextFusionNet	91.8	90.4	87.2	88.5	95.2
SVM	85.4	83.2	78.5	80.8	91.4
Random Forest	86.2	84.3	79.5	81.8	92.1
Naive Bayes	82.7	81.4	76.3	78.7	89.8
LSTM	88.4	86.5	82.9	84.6	93.8
CNN	87.1	85.9	80.8	82.5	92.5

However, the lower recall and F1-score values in the non-SMOTE setting show that even hybrid deep learning models are affected by severe class imbalance. Classical classifiers like SVM, Random Forest, and Naive Bayes perform poorly due to their inability to model high-dimensional spatial-temporal traffic data, and they are more likely to be biased toward the majority class. Based on these results, it is valid to employ SMOTE in the proposed framework, as class balancing is essential to improve sensitivity to minority attack classes and enhance the reliability of IoT intrusion detection.

4.5 Ablation Analysis With SMOTE

Table 5 shows the model performance when the training data is enriched with SMOTE. The results demonstrate that class balancing affects detection performance across all models, especially recall and F1 score, both of which are significant for the IoT ID dataset, which has a class imbalance problem. ContextFusionNet achieves the best performance with 97.2% accuracy, 96.3% precision, 94.6% recall, and 95.4% F1-score. It achieves a higher recall score of 94.6% compared with non-SMOTE (87.2%), and an F1 score of 95.4% compared with non-SMOTE (88.5%). Likewise, the TimeSpaceNet model achieves 85.0% recall and 86.8% F1-score, whereas SMOTE improves recall to 91.0% and F1-score to 92.5%. These results are enough to show that the SMOTE is not only the preprocessing step but also a significant part for enhancing the detection performance for the minority class from the unbalanced IoT data in the cybersecurity scenario.

Table 5: Performance of IoT network attack detection models using the SMOTE algorithm on the IoT Bot dataset.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	ROC-AUC (%)
TimeSpaceNet	95.4	94.2	91.0	92.5	98.0
ContextFusionNet	97.2	96.3	94.6	95.4	99.1
SVM	90.1	88.5	83.2	85.8	94.9
Random Forest	92.2	91.1	87.1	88.6	96.5
Naive Bayes	89.0	87.5	83.1	85.3	93.2
LSTM	93.1	91.7	89.0	90.3	97.6
CNN	92.7	90.8	86.5	88.6	96.0

The comparison in Figs. 6 and 7 further shows that ContextFusionNet consistently outperforms classical machine learning and standalone deep learning baselines. Although models such as SVM, Random Forest, Naive Bayes, CNN, and LSTM also benefit from SMOTE, their improvements remain lower than those of the proposed hybrid models because they do not fully exploit spatial-temporal traffic relationships. Overall, the combination of SMOTE-based balancing, CNN-based spatial extraction, BiLSTM-based temporal modeling, contextual fusion attention, and ADOPT optimization enables ContextFusionNet to achieve stronger and more balanced IoT attack-detection performance.

4.6 Robustness under Different Class-Imbalance Conditions

To further evaluate robustness, the models were examined under different class-imbalance conditions by comparing performance before and after SMOTE-based rebalancing. The non-SMOTE setting represents the original highly skewed attack distribution, while the SMOTE setting represents a more balanced training distribution. The findings indicate that hybrid approaches in both settings outperform the baseline models, with the hybrid approach achieving significantly higher recall and F1 scores after class balancing. This implies that the architecture's power can be used for spatial-temporal learning, and SMOTE further increases the sensitivity of minority attack classes. Future research will involve expansion of this analysis to evaluate the performance of various attack distribution percentages and different imbalance ratios to test better generalization in different traffic conditions in the smart city.

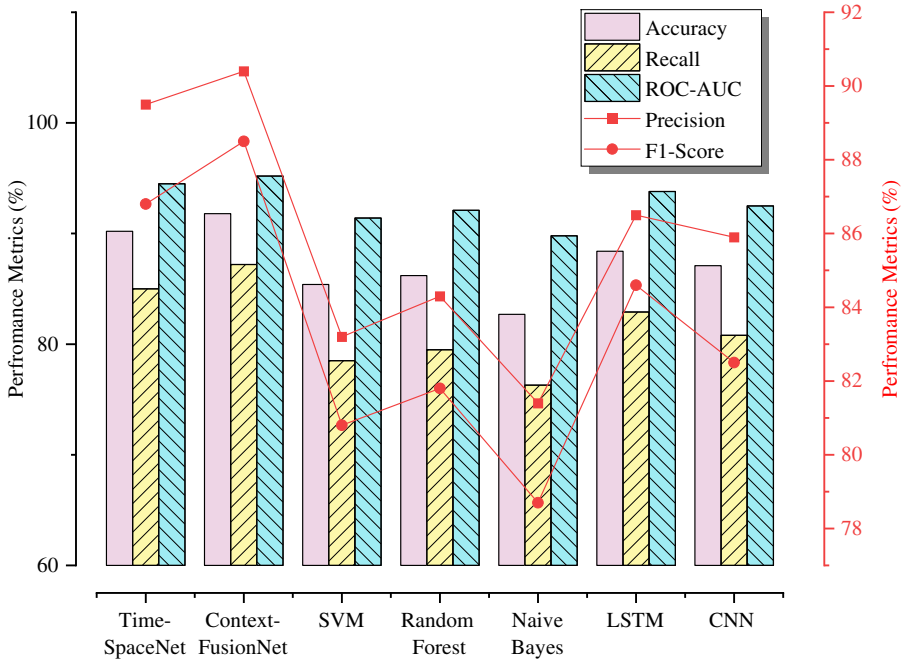

Figure 6: Performance comparison of models without the SMOTE technique.

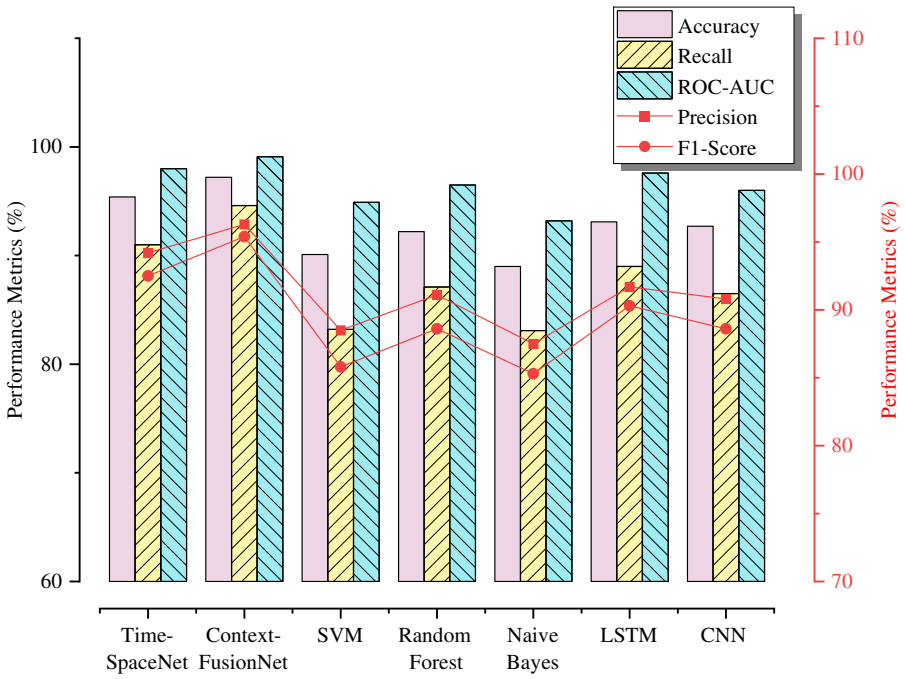

Figure 7: Performance comparison of models using the SMOTE technique.

4.7 Component-Level Ablation Analysis

The component-level ablation analysis reported in Table 6 compares the full ContextFusionNet model with reduced variants. Removing spatial-temporal normalization reduces feature stability between CNN and recurrent layers, which negatively affects convergence and classification consistency. Replacing BiLSTM with a unidirectional LSTM reduces the model's ability to capture bidirectional temporal dependencies, leading to weaker recall for complex attack sequences. Removing contextual fusion attention reduces the model's ability to emphasize discriminative spatial-temporal features, which lowers F1-score and minority-class sensitivity. These findings show that the final ContextFusionNet performance is not caused by a single component, but by the combined contribution of normalization, bidirectional temporal modeling, attention-based fusion, SMOTE balancing, and adaptive optimization.

Table 6: Ablation analysis of ContextFusionNet components and preprocessing strategy.

Model Variant	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Full ContextFusionNet	97.2	96.3	94.6	95.4
Without spatial-temporal normalization	95.8	94.9	92.1	93.4
Without BiLSTM	95.4	94.2	91.0	92.5
Without contextual fusion attention	96.1	95.0	92.8	93.9
Without SMOTE	91.8	90.4	87.2	88.5

4.8 Discussion

The proposed hybrid models were compared with classical machine learning methods, including SVM, Random Forest, and Naive Bayes, as well as standalone deep learning models such as CNN and LSTM. ContextFusionNet achieved the best overall performance, with 97.2% accuracy, 96.3% precision, and

94.6% recall, outperforming TimeSpaceNet and all baseline models. This improvement is mainly due to the combined effect of CNN-based spatial feature extraction, BiLSTM-based bidirectional temporal modeling, contextual fusion attention, SMOTE-based class balancing, and ADOPT optimization. CNN layers capture local traffic-feature relationships, BiLSTM layers model both forward and backward temporal dependencies, and attention fusion emphasizes the most discriminative spatial-temporal features. In contrast, standalone CNN models lack temporal modeling, standalone LSTM models lack explicit spatial extraction, and traditional machine learning models are less effective for high-dimensional sequential IoT traffic.

TimeSpaceNet also worked well, particularly with simpler temporal patterns, but because of its unidirectional LSTM design, it has less appeal for encoding broad context, as does the ContextFusionNet design. With SMOTE, the detection of the minority class was improved, with less influence from high class imbalance, whereas ADOPT led to greater convergence during training. The results appear marginally higher on the computational cost side due to BiLSTM processing and attention fusion, but the increase in recall and F1-score is justified. However, there still needs to be an evaluation of real-world traffic, energy consumption, latency, and memory usage in a real deployment on a resource-constrained IoT and edge device.

5 Conclusion

In this study, two hybrid deep learning models were presented for detecting cyberattacks in IoT environments for smart cities: TimeSpaceNet and ContextFusionNet. TimeSpaceNet also integrates CNN-based Spatial feature extraction, LSTM-based temporal modeling, and Spatial-temporal Normalization, whereas ContextFusionNet adds bidirectional LSTM-based temporal learning and contextual fusion attention. Experiments on the IoT Bot dataset demonstrated that the proposed ContextFusionNet achieved superior overall results compared to TimeSpaceNet and to machine learning and deep learning models, with 97.2% overall accuracy, 96.3% precision, 94.6% recall, and 95.4% F1-score. These results show that combining spatial feature extraction, temporal dependency learning, attention-based fusion, SMOTE-based class balancing, and ADOPT optimization can enhance IoT intrusion detection performance, especially for underrepresented attack classes.

Despite these promising results, the current evaluation is limited to an offline benchmark dataset, and the practical deployment readiness of the proposed models has not yet been fully validated under live IoT traffic, edge-device resource constraints, communication delays, memory limitations, and changing attack distributions. Future work will therefore focus on cross-dataset validation using additional IoT intrusion-detection datasets, real-time edge deployment, latency and computational-cost benchmarking, adversarial robustness analysis, and further optimization for scalable smart-city IoT cybersecurity systems.

Acknowledgement: The authors would like to express their sincere gratitude to the Centre for Research and Innovation Management (CRIM), Universiti Teknikal Malaysia Melaka (UTeM), for its valuable support of this research.

Funding Statement: Not applicable.

Author Contributions: Conceptualization, Muhammad Usman Ghani and Muhammad Javed; methodology, Zeeshan Ali Haider and Inam Ullah; software, Muhammad Usman Ghani and Fida Muhammad Khan; validation, Mohd Faizal Bin Yusof and Jamil Abedalrahim Jamil Alsayaydeh; formal analysis, Zeeshan Ali Haider and Fida Muhammad Khan; investigation, Muhammad Javed and Muhammad Usman Ghani; resources, Mohd Faizal Bin Yusof and Jamil Abedalrahim Jamil Alsayaydeh; data curation, Fida Muhammad Khan and Zeeshan Ali Haider; writing—original draft preparation, Muhammad Usman Ghani and Muhammad Javed; writing—review and editing, Inam Ullah, Jamil Abedalrahim Jamil Alsayaydeh and Mohd Faizal Bin Yusof; visualization, Zeeshan Ali Haider and Fida Muhammad Khan; supervision, Inam Ullah and Jamil Abedalrahim Jamil Alsayaydeh; project administration, Muhammad Javed and Mohd Faizal Bin Yusof. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The BoT-IoT data set is free to academic researchers and has more than 72 million records of the IoT network activity, comprising Botnet, DDoS, Malware, and Port Scanners attacks. Available in PCAP, Argus, and CSV formats, it can be accessed through the UNSW Canberra Research Portal and Kaggle.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhang K, Wang Y, Bhatti UA, Zhou Y, Jin M. Enhanced ransomware attacks detection using feature selection, sensitivity analysis, and optimized hybrid model. J Big Data. 2025;12(1):245. doi:10.1186/s40537-025-01289-1.
2. Zhang K, Wang H, Chen M, Chen X, Liu L, Geng Q, et al. Leveraging machine learning to proactively identify phishing campaigns before they strike. J Big Data. 2025;12(1):124. doi:10.1186/s40537-025-01174-x.
3. Zhou Z, Shojafar M, Alazab M, Abawajy J, Li F. AFED-EF: an energy-efficient VM allocation algorithm for IoT applications in a cloud data center. IEEE Trans Green Commun Netw. 2021;5(2):658–69.
4. Demertzi V, Demertzis S, Demertzis K. An overview of cyber threats, attacks and countermeasures on the primary domains of smart cities. Appl Sci. 2023;13(2):790. doi:10.3390/app13020790.
5. Xiong Z, Luo Q, Zhao G, Zhang X, Chen H, Li Y, et al. A multimodal-based approach for smart contract vulnerability detection. J Signal Process Syst. 2026;98(1):17. doi:10.1007/s11265-026-01985-y.
6. Wang T, Liu M, Li H, Zhao L, Jiang C, Xia C, et al. ArchSentry: enhanced android malware detection via hierarchical semantic extraction. IEEE Trans Netw Serv Manag. 2025;22(3):2822–37.
7. Dinh HM, Zong W, Chow YW. Investigating oversampling techniques to mitigate class imbalance in network intrusion detection datasets. Pragmatic Cybersecur. 2026;1(1):4.
8. Qudus L. Advancing cybersecurity: strategies for mitigating threats in evolving digital and IoT ecosystems. Int Res J Mod Eng Technol Sci. 2025;7(1):3185.
9. Kaur G, Balyan V, Gupta SH. An overview of deep learning techniques for big data IoT applications. Int J Commun Syst. 2026;39(4):e70418. doi:10.1002/dac.70418.
10. Xu G, Xu S, Fan X, Cao Y, Mao Y, Xie Y, et al. RAT ring: event driven publish/subscribe communication protocol for IIoT by report and traceable ring signature. IEEE Trans Ind Inform. 2025;21(9):6670–8.
11. Ribeiro AV, Madureira AL, Sampaio LN. Secure by design: merging network and security bootstrapping for IoT systems through NDN. Comput Netw. 2026;112162.
12. Aldhaheri A, Alwahedi F, Ferrag MA, Battah A. Deep learning for cyber threat detection in IoT networks: a review. Internet Things Cyber-Phys Syst. 2024;4(3):110–28. doi:10.1016/j.iotcps.2023.09.003.
13. Khacha A, Aliouat Z, Harbi Y, Gherbi C, Saadouni R, Harous S. Landscape of learning techniques for intrusion detection system in IoT: a systematic literature review. Comput Electr Eng. 2024;120(7):109725. doi:10.1016/j.compeleceng.2024.109725.
14. Wedamuni Arachchige RN, Saxena D, Singh AK. An adaptive cyber threat intelligence model to counter evolving security attacks in industrial communication networks. Neural Comput Appl. 2026;38(2):15. doi:10.1007/s00521-025-11765-7.
15. Abd Alsadh MH, Abdulateef AN, Al-Amshawi MZ, Taha MA, Najim AH, Ahmed AA, et al. Improving intrusion detection in IoT networks with a hybrid CNN-BiLSTM deep learning model. Int J Intell Eng Syst. 2026;19(1):473.
16. Joloudari JH, Marefat A, Nematollahi MA, Oyelere SS, Hussain S. Effective class-imbalance learning based on SMOTE and convolutional neural networks. Appl Sci. 2023;13(6):4006. doi:10.3390/app13064006.
17. Rathinasamy D, Kaliappan VK. Robust cyber defense framework with Adamax convolutional neural network and hierarchical multi-scale long short-term memory. Transp Res Rec. 2026;2680(1):862–81. doi:10.1177/03611981251362778.
18. Shafin SS, Karmakar GC, Mareels I, Balasubramanian V. Obfuscated memory malware detection in resource-constrained IoT devices for smart city applications. Sensors. 2023;23(11):5348. doi:10.3390/s23115348.