Cluster Computing
https://doi.org/10.1007/s10586-024-04434-2

Cepair: an AI-powered and fog-based predictive CEP system for air quality monitoring

Mehmet Ulvi Şimsek[1] · İbrahim Kök[2] · Suat Özdemir[3]

Received: 12 September 2023 / Revised: 10 March 2024 / Accepted: 13 March 2024

Abstract
Air pollution is one of the influential problems threatening the environment and human health today. Therefore, it is critical to develop predictive systems for proactive decisions in solving this problem. Since the prediction of air pollution depends on several complicated factors such as the accuracy of meteorology reports, air pollution accumulation, traffic flow, and industrial emissions, the contribution of historical or real-time predictions to the solution of the problem is limited. To address the existing limitations, we propose a novel AI-powered and Fog-based predictive complex event processing system (CepAIr) for the prediction of future air pollution rates. CepAIr predicts the future air quality of pollutant gases using RNN, LSTM, CNN, and SVR models. Then, it sends the prediction results to decision-makers in an understandable format, enabling them to take proactive actions. Finally, we evaluate the performance of the CepAIr with SVR and DL models. Additionally, we examine CepAIr in terms of end-to-end network delay and measure its impact on the network. The extensive simulation results demonstrate that the CepAIr predicts future pollutant gas concentrations with DL models (especially with CNN) with a high success rate while guaranteeing minimum end-to-end network delay.

Keyword Air pollution · Deep learning · Internet of things (IoT) · Fog computing · Complex event processing (CEP)

1 Introduction

Air pollution is a serious environmental problem that attracts worldwide attention due to its adverse effects on human health and sustainable development [1]. With the rapid development of industrialization, urbanization, and the economy, many developing countries suffer from heavy air pollution [2]. The main sources of air pollution are particulate matters ($PM_{2.5}$ and PM_{10}) and pollutant gases (O_3, NO_2, SO_2, CO) [3]. A remarkable and constant increase in concentrations of these substances directly reduces air quality and causes the air pollution problem. It also poses a major threat to health, economy, ecology, and climate in the long run [4]. According to the World Health Organization (WHO), seven million people die every year from air pollution. These deaths are largely due to many acute and chronic health issues caused by air pollution such as strokes, heart disease, asthma, obstructive pulmonary disease, lung cancer, and respiratory infection [5]. Air pollution is divided into three categories: indoor, outdoor, and general. It means that the air pollutants can be found at home, in the workplace, and also outdoors [6]. We face increasing levels of air pollution that can be a challenging issue to solve for governments. The limits of the air pollutant gases are exceeded in many megacities that have emerged as a problem and have to be managed by authorities in urban management. To prevent air pollution in cities, many urban managers use predictive system to inform and supply the right information about air pollution to the citizens [7]. This predictive system not only provides air pollution information to citizens but also enables measurements to be made to prevent air pollution.

✉ Mehmet Ulvi Şimsek
mehmetulvi@gmail.com

İbrahim Kök
ikok@pau.edu.tr

Suat Özdemir
ozdemir@cs.hacettepe.edu.tr

[1] ASELSAN, Ankara, Turkey

[2] Department of Computer Engineering, Pamukkale University, Denizli, Turkey

[3] Department of Computer Engineering, Hacettepe University, Ankara, Turkey

Published online: 16 April 2024

This information shows that air pollution is one of the major challenges for governments and decision-makers to protect human health and the environment. Therefore, it is necessary to predict the pollutant gas concentrations and keep the air quality at an appropriate level in order to take precautions against air pollution [8].

Air pollution prediction has attracted the attention of researchers for years. Traditional methods are applied to solve the air pollution prediction problem. The traditional methods use statistical and mathematical methods. However, it has some deficiencies as limited accuracy, cut-offs and complex mathematical calculations [9]. On the other hand, the conventional methods use data mining algorithms for predicting air pollutants that use regression, classification, clustering and association mining algorithms [6]. There is a huge attention to artificial intelligence by researchers that use these algorithms to predict air pollutants. AI models use artificial neural networks, support vector machines and other deep learning algorithms. Due to the limitations of these algorithms, hybrid approaches are proposed to solve the problem of air pollutant prediction. [10]. Moreover, big data analytics models are used for air pollutant prediction with better accuracy results [9].

Complex Event Processing (CEP) is a framework that is used for detecting complex events from atomic events in real time. With the help of real-time operations, CEP is capable of detecting emerging issues about the given rules from the CEP framework. The atomic events can be represented as a stream of sensors that come together to form complex events. The complex events are detected via predefined rules which are mostly defined by domain experts [11].

However, both predicting pollutant gas concentrations and evaluating prediction results are complex processes. The complexity of the process arises from the real-time processing, analysis and correlation of data streams from different gas sensors [12]. Therefore, there is a need for a sophisticated system to detect complex events in real-time data streams [13, 14]. At this point, Event-Driven Architectures (EDAs), which enable the processing of events in the data stream at the time they are received and with minimum delay, have an effective solution potential [15]. CEP system that lies under EDAs allows us to process and analyze a large number of heterogeneous data streams to detect relevant or critical situations for a particular domain. In this way, it becomes possible to extract events in real-time and evaluate their consequences. On the other hand, CEP systems lack the predictive power of machine learning and statistical data analysis methods [16]. For this reason, the CEP system alone is insufficient in developing a predictive environmental impact assessment system and early warning system that can be a solution to air pollution [17, 18].

In this context, the main motivation of this paper is to present a conceptual CEP system model for air pollution early warning system combining Deep Learning (DL) and traditional machine learning methods. In this system model, we synthesize the predictive power of DL models and the effective data and event processing power of the CEP system. Herein, we design a system structure consisting of the prediction unit, the rule engine, the CEP engine and the decision unit components. Then, we deploy the designed system in accordance with the fog computing concept. With its aforementioned innovative features, the proposed system is the first study to jointly address DL and CEP concepts in solving the air pollution problem.

The main contributions of this paper are summarized as follows

- We propose a novel predictive CEP system (CepAIr) that detects air pollution events by predicting the future values of air pollutants real time. In CepAIr, each pollutant concentration is predicted separately, and simple/complex pollution events are determined by jointly evaluating the future concentration levels.
- We propose LSTM, RNN, and CNN-based DL models to predict the future concentration levels of six air pollutants ($PM_{2.5}$, PM_{10}, O_3, NO_2, SO_2, CO) in the predictive unit of the CepAIr.
- We present a system structure that detects comprehensive future air pollution events by combining the strengths of the proposed DL models and the proposed CEP engine. The system conveys the alarm level of detected events to decision-makers in an understandable format.

The remainder of this paper is organized as follows. Section 2 briefly presents related works and background. Section 3 introduces the details of the proposed AI-powered and fog-based CEP (CepAIr) system. Section 4 presents the experimental results and performance evaluations. Finally, Sect. 5 concludes the paper.

2 Background and related works

In this subsection, we present related works, background information, and abbreviations (see Table 1) used in the paper.

2.1 Related works

In recent years, DL has drawn widespread attention in many domain-specific applications. In the literature, although many studies predict air quality based on DL and statistical methods, the issue of predictive CEP is not addressed in these studies. Therefore, in this section, we

Table 1 List of abbreviations

Abbreviations	Definition
AMWR	Adaptive Moving Window Regression
AQI	Air Quality Index
CEP	Complex Event Processing
CNN	Convolutional Neural Network
CO	Carbon Monoxide
DL	Deep Learning
DT	Decision Tree
EDA	Event-Driven Architecture
EM	Expectation Maximization
EPA	Environmental Protection Agency
GMM	Gaussian Mixture Model
IoT	Internet of Things
LR	Logistic Regression
LSTM	Long Short Term Memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
ML	Machine Learning
MLP	Multilayer Perceptron
MSE	Mean Squared Error
NB	Naive Bayes
NO_2	Nitrogen Dioxide
O_3	Ozone
PCPN	Prioritized Colored Petri Net
PM	Particule Matter
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
SO_2	Sulfur Dioxide
SVM	Support Vector Machine
SVR	Support Vector Regressor
SWR	Semantic Web Rule
VEQL	Video Event Query Language
WHO	World Health Organization

summarize current papers that examine traditional machine learning and CEP system together in different fields.

Fülöp et al. [19] propose a conceptual framework to show efficiency of the combining predictive analysis and the CEP system. The framework uses BFtree, decision tree and J48 for predictive analyses. Furthermore, the prediction performance is evaluated in terms of precision, recall, and F-measure metrics Schwegmann et al. [20] propose an architecture to combine predictive process analytics and CEP. The authors design a predictor unit that uses regression and classification algorithms. The prediction unit, which acts as an event producer, make prediction with the predictor metrics. They evaluate the architecture in terms of MSE and classification accuracy metrics. Cabanillas et al. [21] propose a framework that integrates predictive analysis with CEP in business process management. The authors employ the SVM algorithm on air flight data by integrating Esper CEP. Wang et al. [22] propose a method that uses the Bayesian network to predict future events by integrating probabilistic complex event processing components into architecture. Similarly, Wang et al.[23] propose a predictive event processing agent to send predictive complex events to decision support systems. They employ the Bayesian model to predict future events. The proposed model shows that accuracy values increase when the window size increases in the road traffic domain. Nechifor et al. [24] propose a framework that predicts the temperature of parcels in the supply chains. In the framework, the authors use time series prediction in the CEP system with a multilayer perceptron model. However, the evaluation of the prediction step is not clearly given in the paper. Christ et al.[25] present an architecture to integrate predictive analysis component within the CEP system. In this study, the predictive component feeds the CEP engine using Conditional Density Estimation algorithms. However, the prediction performance is not evaluated. Akbar et al. [12] present an architecture that performs accurate predictions in CEP system for IoT data. In this study, they combine the power of real-time and historical data processing using CEP and ML. Furthermore, they use an adaptive moving window regression algorithm called (AMWR) for prediction. The architecture provides early warnings for traffic events with high accuracy on traffic dataset. In [26], the authors employ Logistic Regression and Naive Bayes to predict the network fault in CEP-PA system. They evaluate the model in terms of prediction metrics. Xing et al. [27] propose a framework that integrates the concepts of deep learning models with complex event processing engines. Furthermore, the authors use DL to tag the primitive events which provide semantic meaning on a video stream.

Yadav et al. [28] propose a framework to predict traffic using deep learning techniques on camera video streams. Furthermore, the framework is evaluated in terms of real-time latency and F-score metrics. The results show that multidimensional analyses are made by integrating the DL methods to CEP architecture.

Diaz et al. [29] propose an Intelligent Transportation System model based on Complex Event Processing and Colored Petri Nets (CPNs) to make decisions about traffic regulations to reduce air pollution levels in large cities. In the same context, the authors in another study [30] propose an approach, MEdit4CEP-CPN, which extends the Prioritized Colored Petri Net (PCPN) formalism to support the modeling, simulation, analysis, and both syntactic and semantic validation of complex event-based systems. The suggested approach has been validated through a case study on air quality level detection to demonstrate its

Table 2 Summary of related works

References	Application Domain	Research Focus/Problem	Used Method/Technique	Query Language
Schwegmann et al. [20]	Business	Predictive Process Analytics	DT, SVM, Rule model, Regression	NA
Cabanillas et al. [21]	Business	Process Management	SVM	Esper - EPL
Christ et al [25]	Business	Process Management	Conditional Density Estimation	Rule-based
Liu et al. [34]	Energy	Air Pollution	LSTM, DT	Custom
Semsali et al. [31]	Environment Monitoring	Air Pollution	NA	Esper - EPL
Fülöp et al. [19]	Generic	Predictive Analytics	Btree, DT, J48	Esper - EPL
Yemson et al. [35]	Healthcare	Air Quality	LSTM, BiLSTM, GRU	SWR
Wang et al. [22]	Intelligent Transportation System	Proactive Event Processing	Markov Decision Process (MDP)	NA
Diaz et al. [29]	Intelligent Transportation System	Air Pollution	Colored Petri Nets (CPNs)	Esper - EPL
Diaz et al. [30]	Intelligent Transportation System	Air Quality	Prioritized Colored Petri Net (PCPN)	Esper - EPL
Nechifor et al. [24]	Logistic	Cold Chain Monitoring	MLP, ARIMA	NA
Xing et al [27]	Physical Security	Human Activity Recognition	YOLOv3	Custom
Akbar et al. [12]	Road Traffic	Predictive Analytics	Adaptive Moving Window Regression	Esper - EPL
Macia et al. [33]	Smart City	Air Quality	Fuzzy Logic	Rule-based
Emerson et al. [26]	Telecommunication	Fault Prediction	Logistic Regression, Naive Bayes	NA
Brazalez et al. [32]	Traffic, Industry, Domestic, Agricultural	Air Pollution	Fuzzy Logic	Esper - EPL
Wang et al. [23]	Transportation/Road Traffic	Predictive Event Processing	Bayesian Network (GMM and EM)	NA
Yadav et al. [28]	Transportation/Road Traffic	Traffic Prediction	DNN, YOLOv3	VEQL

NA not available

benefits in the modeling, simulation, analysis, and semantic validation of complex event-based systems. Semsali et al. [31] have developed a software architecture called SAT-CEP-Monitor that supports decision-makers by performing complex event processing on remote sensing data based on satellite sensors. The architecture has been validated on several regions of Morocco and Spain based on ground station and satellite data. Brazález et al. [32] have proposed a decision support system named FUME that can provide daily recommendations based on reference air pollution standards. FUME aims to improve the decision-making process by suggesting actions for pollution sources in traffic, industrial, domestic, and agricultural areas, utilizing fuzzy logic and CEP technologies. In FUME, CEP conducts real-time analysis based on information collected from different stations and weather forecast data, providing notifications accordingly. Fuzzy logic, on the other hand, has facilitated working with linguistic variables and uncertain data by relying on the knowledge of the domain expert. In a similar study, Macia et al. [33] have proposed a methodology supporting the action plan for air pollution in cities based on Complex Event Processing (CEP) and Fuzzy Logic. Liu et al. [34] have introduced a new CEP rule auto-extraction framework named LAD, which combines a two-layer LSTM with attention mechanism and decision tree data mining approach. In this framework, the authors leverage deep learning in two phases: filtering and labeling abnormal data, and focusing on the extraction of pattern rules with high accuracy. Experimental results showed that the framework effectively extracts meaningful CEP rules, supports real-time air quality monitoring, and contributes to air pollution prediction and regulatory strategies. Yemson et al. [35] proposed a CEP framework for indoor air quality prediction based on ontologies in the semantic web domain. The proposed framework takes inputs of particulate matter, carbon dioxide, humidity and temperature and provides notification and warning for complex event and anomaly detection. In the study, a rule-based approach is used for complex event and anomaly detection.

In the existing literature, predictive CEP has found applications in various fields, but there is a noticeable gap in its exploration in the IoT environmental monitoring domain. Moreover, widespread studies are predominantly based on traditional ML models. In this context, it becomes necessary to incorporate advanced deep learning models to achieve high accuracy levels in CEP systems. In addition, the architectural and computing aspects of the proposed CEP systems are largely unaddressed. This paper aims to address these shortcomings by presenting a novel CEP system based on DL models and utilizing a fog computing approach for air pollution monitoring. This study aims to contribute to the existing body of knowledge by reducing the identified gaps in the existing literature (Table 2).

2.2 Background

In this subsection, we provide background information on the DL models used in this study.

2.2.1 Recurrent neural network (RNN)

RNN has a simple structure including input, output and hidden layers. One of the prominent achievements of RNN is its superiority in time convergence [36]. Besides, it can easily learn the temporal dependencies with its simpler structure. The RNN structure can be mathematically formulated as follows:

$$h(t) = f_H(W_{IH}x(t) + W_{HH}h(t-1)) \tag{1}$$

$$y(t) = f_O(W_{HO}h(t)) \tag{2}$$

where $x(t)$ is a sequence of time series data. f_H and f_O are the hidden and output unit activation functions. $h(t)$ is defined as a hidden layer. It is computed by input x and previous hidden layer $h(t-1)$. W_{IH}, W_{HH}, and W_{HO} are the weight matrices. y is an output value which is calculated by Eq. 2.

2.2.2 Long short term memory (LSTM)

LSTM is a special Recurrent Neural Network (RNN) structure that was proposed for learning long-term dependencies by Hochreiter and Schmidhuber [37]. The main innovation of the LSTM is a memory cell that preserves the state of information in a hierarchical structure [38, 39]. Furthermore, the memory blocks contain input, output, and forget gates that regulate the flow of information from cell to cell [40]. The LSTM blocks can be formulated as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{3}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{4}$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{5}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{6}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{7}$$

$$h_t = o_t * \tanh C_t \tag{8}$$

where x_t is input vector and h_t output vector. t is expressed as a time. $\tilde{C}_t$ and C_t are the old and new cell state. i_t,f_t and o_t are the input, forge,t and output gates respectively, respectively. W_f, W_i, W_c, W_o are the input weights matrices, b_f, b_i, b_c, b_o are the bias vectors. $\sigma(\cdot)$ is a the logistic sigmoid function, i.e $\sigma(x) = \frac{1}{1+e^{-x}}$ and $\tanh(\cdot)$ are the hyperbolic tangent function.

2.2.3 Convolutional neural network (CNN)

CNN is popular with image recognition and is used as a powerful technique in spatial data processing [41]. There are three types of dimensionality in CNN architecture including one, two, and three-dimensional. CNN one-dimensional is mostly used in time series data. Because the vector of the time series can be represented as one-dimensional. On the other hand, CNN two and three-dimensional are used image or video data because the image has two or three-dimensional [42]. CNN has two layers including pooling and convolutional. In the pooling layer, the input or out of the previous layer dimensions is reduced which is called subsampling. In the convolutional layer, smaller feature maps are inferred from the input data with the help of the sliding process of convolutional matrix [43, 44].

The calculation methods of the CNN are represented as follows [45].

$$w_{out} = (W_i n - F)/s + 1 \tag{9}$$

$$w_{out} = (W_i n + 2 * p - F)/s + 1 \tag{10}$$

$$Sigmoidfunction : f(z) = 1/(1 + e^{-z}) \tag{11}$$

where w_{in} and w_{out} are input and output feature map, F is the convolution kernel size, s represents the convolution step size, and p is expressed as a the number of pixels.

2.2.4 Support vector regression (SVR)

SVR is a special type of support vector classification [46]. It is used as a regression technique in many problems[47] that the algorithm is used as predicting intervals. Furthermore, it uses statistical learning theory to minimize error [48]. The regression formula is formulated as follows [49].

$$f(x) = (W, O(x)) + b \tag{12}$$

where w and b are regression parameters, x is represented as a vector of input, O is the kernel operator.

3 Proposed AI-powered and fog-based predictive CEP system (CepAIr)

In this section, we explain our network model and formulation. Then, we present our AI-powered and Fog-based Predictive CEP system (CepAIr), the proposed DL models, and optimization parameters.

3.1 Network model and formulation

As shown in Fig 1, we consider a fog-based IoT network that consists of physical devices (pollution sensing sensors, clients, end devices, etc.), fog server and cloud servers. We assume that the sensors in the physical layer are placed in groups in different geographic regions. In network model, P is the set of n pollutant sensors for each region (R), where $P = \{p_1, p_2, ..., p_n\}, R = \{r_1, r_2, ..., r_n\}$. l_{ef}^{up}, l_{fc}^{up} are the uplinks and l_{cf}^{down}, l_{fe}^{down} are the downlinks between the layers, where e is physical device, f is fog device. We use D_r^{down} to denote the input data size that is transmitted from fog/cloud servers to the physical devices. We use D_r^{up} to denote the output data size of n pollutant sensors. In addition B^{up} and B^{down} denote the total uplink bandwidth and downlink bandwidth respectively.

Based on the notations above, we formulate the computational delay and transmission delays below to evaluate the performance of the CepAIr in the network model. Therefore, uplink and downlink transmission delays can be given as shown in Eqs. 13 and 14.

$$T_{l_{ef}/l_{fc}}^{up} = \frac{D_r^{up}}{B^{up}} \tag{13}$$

$$T_{l_{cf}/l_{fe}}^{down} = \frac{D_r^{down}}{B^{down}} \tag{14}$$

Thus, we can get an approximate end-to-end transmission delay (T_{e2e}^{trans}) for each packet by summing all uplink and downlink transmission delays.

$$T_{e2e}^{trans} = T_{l_{e,f}}^{up} + T_{l_{f,c}}^{up} + T_{l_{e,f}}^{down} + T_{l_{f,c}}^{down} \tag{15}$$

Next, we model the CepAIr processing delay (T_{CepAIr}^{proc}). Herein, we assume the delay times in the prediction unit, CEP engine, and decision unit as T_{pu}^{pred}, T_{ce}^{proc} and T_{du}^{proc}, respectively. Specifically, the prediction delay of the prediction unit can be given by

$$T_{pu}^{pred} = \frac{D_r^{up}}{f^{cc}} \tag{16}$$

where $f^{cc}$is the computing capability of fog server. Thus, we obtain the CepAIr processing delay by summing up the prediction and processing delays in all system units.

$$T_{CepAIr}^{proc} = T_{pu}^{pred} + T_{ce}^{proc} + T_{du}^{proc} \tag{17}$$

Finally, the total network delay for N packets transmitted to the network from each region can be given as follows.

$$T_{total}^{network} = \sum_{n=1}^{N} (T_{CepAIr}^{proc} + T_{e2e}^{trans})$$

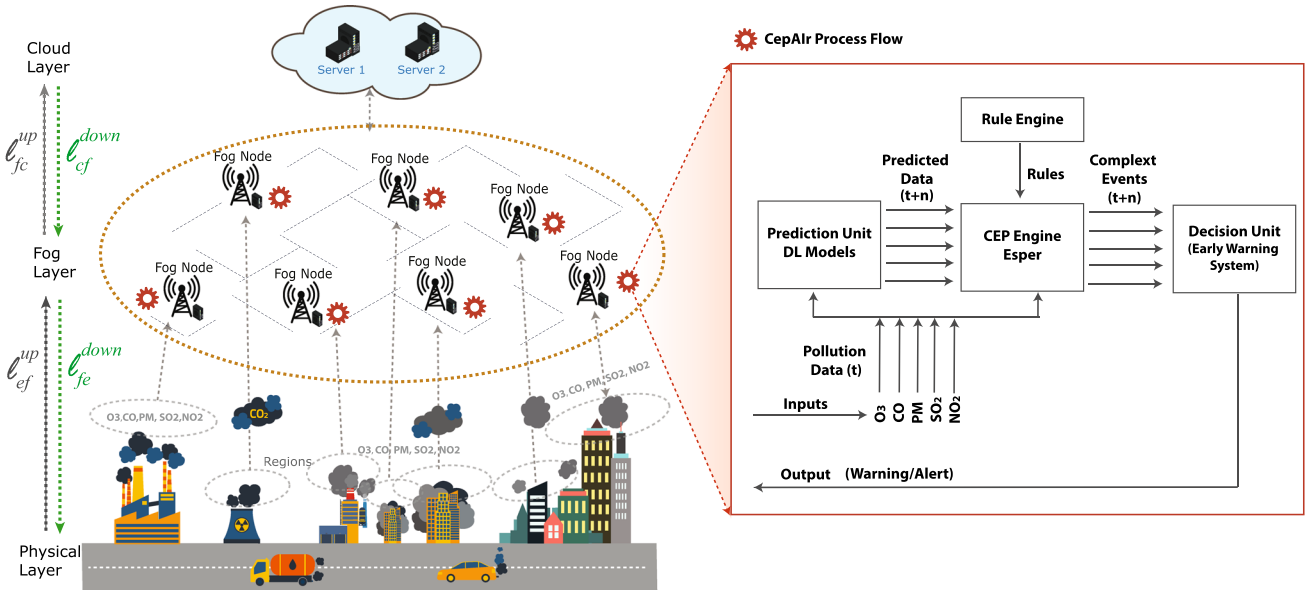

Fig. 1 Overall Network model (Left) and CepAIr Process Flow (Right)

3.2 Proposed CepAIr system

In this paper, we address the air pollution problem that threatens human health as a major problem worldwide. It is known that this problem varies according to the country, city and even specific location. Therefore, potential solutions to this problem should be location-based and distributed due to the nature of the problem. Based on this information, we perform air pollution monitoring, analysis and prediction processes based on fog computing approach, which provides the opportunity to perform distributed, fast and effective. Therefore, we propose a system called CepAIr, which includes a prediction unit, a rule engine, a CEP engine, and a decision unit. Then, we explain the functions of these units and the position of the system in the IoT architecture in detail below.

Specifically, CepAIr provides an architectural concept for dealing with air pollution in smart cities using CEP. This system concept is designed according to the fog computing approach in a three-tiered IoT architecture. In the physical layer, the concentration values of ozone, particulate matter, carbon monoxide, sulfur dioxide, and nitrogen dioxide gases that cause air pollution are detected and sent to the closest fog server in the coverage area. In the fog layer, the proposed CepAIr system is distributed over fog servers. Fog servers process the incoming data and put it into the CepAIr system. CepAIr performs future prediction of local air quality by using the prediction values of each pollutant gas. Then it sends the results to the clients directly or via the cloud in interpretable form. As shown in Fig. 1, the proposed CepAIr consists of four major components, namely the prediction unit, CEP engine, rule engine, and decision unit. These components are comprehensively explained below.

- Prediction Unit: This unit predicts future pollution concentrations for each gas using developed DL models and SVR. The models take each gas value and predict the next gas values. Then, it sends the predicted pollution values to the CEP engine. In this unit, future values of pollutant gases are predicted through the developed RNN, LSTM, CNN, and SVR models. The best algorithm is selected after evaluating by prediction performance in each other.
- Rule Engine: The rule engine determines the class rules used in the detection of events according to the expert guides or referenced standards and sends them to the CEP engine. In our case, the rule engine creates many simple and complex rules regarding air pollution according to the US Environmental Protection Agency (EPA) reference AQI values and classes [50] shown in Fig 2.

Air Quality Index (AQI) Values	Levels of Health Concern	Colors
0 to 50	Good	Green
51 to 100	Moderate	Yellow
101 to 150	Unhealthy for Sensitive Groups	Orange
151 to 200	Unhealthy	Red
201 to 300	Very Unhealthy	Purple
301 to 500	Hazardous	Maroon

Fig. 2 EPA air quality index levels [50]

Based on this reference AQI values, our rule engine provides the rules for the three main health concern levels shown in Table 3. More specifically, the rule engine considers 0-50, 51-100, and 101-500 AQI values for good, moderate, and unhealthy levels, respectively. The rules as named normal, warning, and critical are deployed to the CEP engine to detect complex events.

- CEP Engine: CEP engine processes and analyzes large amounts of pollutant gas data by using the rules created by rule engine to detect simple, complex, and composite events. Specifically, the CEP engine in CepAIr provides the detection of simple and complex events for future air pollution by taking the pollutant gas values predicted by DL models and predefined rules in the rule engine. Then it sends the detected events to the decision unit. To perform all these operations, we use the ESPER framework [51] that you can build CEP engine with Java coding and it can be embedded into Java applications.
- Decision Unit: The decision unit evaluates all events and functions as an early warning system. It categorizes and labels the alarm level of all events as Normal, Warning, and Critical in advance and sends it to decision-makers in an understandable format.

To better understand the process flow of the proposed system and the functions of the units given above, we present the whole process flow of DeepFog CEP step by step in Algorithm 1. To explain in detail, the first step is collecting data from sources. In particular, the data is air pollution data which can be deployed in different areas in the urban area. After the data collection step, the pollutant gases next point is predicted via the selected best algorithm comparison to each other. The next step is expressed as getting rules from Rule Engines which is defined as taking care of the level of health concern. Next, The CEP engine detects the complex events via predefined rules and sends the complex events to the decision unit. Lastly, Future alarm levels are determined and sent to the decision-

Table 3 Used AQI levels and rule formats

AQI values	Level of health concern	Alarm label	CEP rule format
0-50	Good	Normal	Select * from AirQualityEvent match recognize (measures A as temp1"pattern (A) define A as A.pollutantGas $\leq$ 50
51-100	Moderate	Warning	Select * from AirQualityEvent match recognize (measures A as temp1"pattern (A) define A as A.pollutantGas>50 and A.pollutantGas $\leq$ 100)
101-500	Unhealty (Hazardous)	Critical	Select * from AirQualityEvent match recognize (measures A as temp1"pattern (A) define A as A.pollutantGas>100)

makers to take precautions or inform to the citizens in smart cities.

Algorithm 1 CepAIr process flow

Input: PollutantData(O_3, CO, PM, SO_2, NO_2), Prediction Models (M_{CNN}, M_{LSTM}, M_{RNN}, M_{SVR}), Rules, AQI Values
Output: Alarm Level of the Air Pollutants (A_{level})
Step 1: Take regional pollution data from sensors in the coverage area
Step 2: Predict future concentrations of each pollutant gas in the Prediction Unit (p^t= model$_x$.predict(O_3, CO, PM, SO_2, NO_2))
Step 3: Get rules (R) from Rules Engine
Step 4: Detect simple and complex event via rules (R) in the CEP Engine (Events=Esper(R, p^t)) and send the Events to the Decision Unit
Step 5: Determine future alarm levels of the events in Decision Unit (A_{level}=DecisionUnit(Events, AQI)) and send A_{level} to decision makers

3.3 Proposed DL models, optimization and parameters

We tuned the all used models with different parameters to get more accurate results. Numerous experiments are conducted to get the best prediction performance. In deep learning algorithms, we adopt many to one strategy to predict the ozone, particulate matter, sulfur dioxide, and nitrogen dioxide. The model takes sensor values as inputs and predicts the next sensor value. We have tested the RNN and LSTM models under different numbers of layers as 1,2,3 and hidden units as 128, 256, and 512. We have also tested different numbers of epochs sizes as 50, 100, 200, 300, 400, 500. We adopt the dropout regularization technique to strengthen the prediction accuracy of the models. In CNN, we have tested different numbers of filters as 128, 256, and 512, and different numbers of epochs size as 50, 100, 200, 300, 400, 500. Similarly, SVR takes each sensor's values to predict the next value. We adopt the grid search technique to find the optimum hyper-parameters. Therefore, the hyper-parameters which are shown in Table 4 are obtained from numerous experiments to build the best structure.

Table 4 Model hyper-parameters

Models	Parameters
CNN	Epoch: 100, Filter size: 256, Max pooling
LSTM	Epoch: 100, Hidden Layers:1, Batch Size: 6, Units:128
RNN	Epoch: 100, Hidden Layers:1, Batch Size: 6, Units: 128
SVR	C=1.0, Epsilon=0.2

In the prediction unit, we used the TensorFlow [52], Keras framework [53], and Sklearn [54] to implement all algorithms. In the CEP engine, We use the eclipse platform to use ESPER CEP library [55] which is written in Java. Thanks to these frameworks, the models are trained offline on a desktop computer equipped with an Intel i7-7700HQ CPU, 16-GB RAM, and GTX 1060TI GPU with 6 GB RAM.

4 Experimental results

4.1 Dataset

In this study, we use the air pollution gases collected in the City Pulse EU FP7 Project [56]. The dataset consists of eight features including nitrogen dioxide, ozone, particulate matter, sulfur dioxide, carbon monoxide, longitude, latitude, and timestamp [57]. Furthermore, the data stream of each gas is continuous and timely annotated and collected at five-minute intervals for 60 days. Table 5 presents the descriptive statistics of the polluted gases.

Additionally, some information on the pollutant gases used in this data set and their effects on human health are presented below [58].

Table 5 Statistical description of the Dataset

Parameters	Ozone	Particule matter	Carbon monoxide	Sulfur dioxide	Nitrogen dioxide
Sample Size	17568	17568	17568	17568	17568
Min	15	15	15	15	15
Max	215	215	215	215	215
Mean	111.04	124.90	98.13	116.59	107.100
Standart Deviation	55.04	54.04	49.70	54.61	54.09

4.1.1 Ozone (O_3)

O_3 is a pollutant that increases as a result of chemical reactions between sunlight, nitrogen oxides from vehicle emissions and volatile organic compounds. High ozone levels can irritate the respiratory tract, increase asthma symptoms, adversely affect lung function, affect the cardiovascular system and contribute to general health problems.

4.1.2 Particule matter (PM)

PM consists mainly of primary pollutants from smokestacks, construction sites, fires, or volcanoes and secondary pollutants from power plants, factories, and vehicles. PM is respirable and can accumulate in the lungs, blocking the airways, increasing respiratory problems and increasing the risk of heart disease.

4.1.3 Carbon monoxide(CO)

CO is an odorless, invisible gas produced by incomplete combustion. The most important sources of this gas are automobile emissions, fires, and industrial processes. Exposure to carbon monoxide in high concentrations and for long periods can lead to serious health problems such as chest pain, vision problems, and reduced physical and mental abilities.

4.1.4 Sulfur dioxide(SO_2)

SO_2 is a colorless pollutant gas with a suffocating odor emitted from sources such as electricity generation, fossil fuel combustion, industrial processes, and automobile emissions. In terms of health effects, SO_2 can cause inflammation in the respiratory tract, increasing respiratory problems and damaging the cardiovascular system.

4.1.5 Nitrogen dioxide(NO_2)

NO_2 is a pungent and irritating pollutant gas that is produced by automobile emissions, electricity generation and industrial processes. Prolonged exposure to NO_2 can cause respiratory symptoms (such as coughing, wheezing or difficulty breathing).

4.2 Performance evaluation of the prediction models

In the prediction unit, we employ four algorithms including SVR, LSTM, RNN, and CNN. We evaluate these prediction models that were developed for use in the prediction unit in terms of mean square error(MSE), root mean square error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), and R-squared (R^2) metrics[59]. The mathematical formulas for the evaluation metrics are given The mathematical formulations of RMSE and MAE metrics are given in Eqs. 18–22.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{18}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{19}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{20}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100 \tag{21}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{22}$$

In these equations, n represents the number of data points, y_i represents the true value, and $\hat{y}_i$ represents the predicted value. The overall average results of the prediction performance are presented in Table 6.

The prediction results of the NO_2, CNN is slightly better than the LSTM, RNN models. On the other hand, SVR in test evaluation shows worse prediction performance in comparison to the other models. According to the O_3 prediction results, CNN, LSTM, and RNN models show similar prediction results when evaluating R^2. The other RMSE, MAE, and MAPE metrics results show the CNN performed better in predicting in compare to the other models. When we evaluate the PM prediction results, it is observed that CNN gives better results, when we evaluate

Table 6 Performance comparisons of the prediction models

Features	Models	Train MSE	Test MSE	Train RMSE	Test RMSE	Train MAE	Test MAE	Train MAPE	Test MAPE	Train R^2	Test R^2
NO_2	SVR	262.91	172.66	16.21	13.14	13.99	10.83	0.20	0.21	0.92	0.86
	RNN	185,23	74,87	13,61	8,65	11,82	7,16	0,14	0,12	0,95	0,94
	LSTM	176,07	56,52	13,27	7,52	11,32	6,23	0,12	0,11	0,95	0,95
	CNN	137,45	35,22	11,72	5,93	9,53	4,87	0,09	0,07	**0,96**	**0,97**
O_3	SVR	306.67	384.09	17.51	19.60	15.00	17.52	0.19	0.41	0.89	0.79
	RNN	70,33	110,32	8,39	10,5	6,92	9,15	0,10	0,22	0,97	0,94
	LSTM	64,93	83,71	8,06	9,15	6,70	7,98	0,09	0,19	0,98	0,95
	CNN	42,05	43,89	6,48	6,63	5,42	5,72	0,06	0,11	**0,98**	**0,98**
PM	SVR	348.98	361.81	18.68	19.02	16.53	17.24	0.19	0.25	0.88	0.88
	RNN	189,42	212,12	13,76	14,56	12,64	13,43	0,16	0,21	0,93	0,93
	LSTM	124,76	145,30	11,17	12,05	9,44	10,24	0,13	0,18	0,96	0,95
	CNN	99,24	102,28	9,96	10,11	9,05	9,31	0,11	0,13	**0,96**	**0,97**
SO_2	SVR	327.11	262.21	18.09	16.19	15.50	13.91	0.27	0.11	0.90	0.87
	RNN	73,45	46,08	8,57	6,79	7,30	5,63	0,14	0,06	0,98	0,98
	LSTM	94,14	59,36	9,70	7,70	8,39	6,45	0,16	0,07	0,97	0,97
	CNN	50,08	38,56	7,08	6,21	6,16	5,25	0,10	0,05	**0,98**	**0,98**
CO_2	SVR	330.10	349.94	18.17	18.71	16.09	17.02	0.25	0.32	0.87	0.81
	RNN	251,53	287,38	15,86	16,95	13,34	15,08	0,23	0,29	0,90	0,85
	LSTM	130,76	159,24	11,44	12,62	9,63	11,36	0,16	0,22	0,95	0,91
	CNN	110,21	3,4	10,50	8,5	8,11	6,64	0,09	0,10	**0,96**	**0,96**

Bold values highlight the best results

all metrics. The SVR model showed lower prediction performance compared to other models. When we consider to the SO_2 prediction results, CNN outperforms the other models. But, RNN and LSTM models show very close performance in the prediction results. On the other hand, SVR model shows the worst performance in comparison to the other models. Lastly, when evaluating CO prediction results in terms of all metrics given in Table, CNN performs better, followed by LSTM, RNN and SVR models.

As seen in Table, the most successful models in terms of all evaluation metrics are CNN, LSTM, RNN, and SVR, respectively. Here, it is seen that CNN models stand out significantly from the other models. This is because CNN can create higher-level representations of sequence data and capture spatial feature dimensions. Based on these results, we use the predicted data obtained by CNN algorithms for detecting future complex events in the CEP engine.

4.3 Classification results of the prediction models for each pollutant gas

In this subsection, we present the classification results of alarm levels in terms of prediction metrics according to the prediction results of the prediction unit. Thus, we demonstrate the success of the decisions made regarding air pollution in the decision unit. The comparative classification results for all gases are presented in Tables 7, 8, 9, 10 and 11. From the comparison of all algorithms in Table 7, we can see that CNN outperforms the other algorithms in comparison to all rule types.

According to Table 7, these scoresshown that CNN-based model outputs have a very good performance in detecting complex events for all rule types as normal, warning, and critical. Then, LSTM, RNN, and SVR models follow the CNN model in order of success.

According to Table 8, experimental results reveal that the CNN algorithm shows above 90% classification performance, except normal rule types, in terms of F1-score and accuracy metrics. However, the SVR algorithm shows underperformance when compared to the other algorithms.

Next, we give the results of the particulate matter gases in Table 9, according to this table, it is seen that all algorithms achieve more than 90% classification performance for critical rule types. However, the CNN model performed better for normal and warning rule types. This result shows that the predicted values of the CNN model achieve a higher classification result in the CEP engine.

According to Table 10, all algorithms show above 94% classification performance for critical rule types. However,

Table 7 The classification results of the nitrogen dioxide (NO_2)

	CNN			LSTM			RNN			SVR		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
Normal	1,000	0.741	0.852	1,000	0.502	0.669	1,000	0.103	0.186	0	0	0
Warning	0.891	0.988	0.937	0.835	0.983	0.903	0.747	0.955	0.838	0.714	0.879	0.788
Critical	0.984	0.959	0.972	0.977	0.973	0.975	0.942	0.987	0.964	0.859	0.996	0.923
Avarage	0.942	0.937	0.935	0.913	0.899	0.889	0.747	0.955	0.838	0.646	0.774	0.704

Table 8 The classification results of the ozone (O_3)

	CNN			LSTM			RNN			SVR		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
Normal	1,000	0.774	0.852	1,000	0.503	0.670	1,000	0.184	0.311	0	0	0
Warning	0.891	0.988	0.937	0.739	0.949	0.831	0.627	0.928	0.749	0.562	0.866	0.681
Critical	0.984	0.959	0.972	0.920	1,000	0.958	0.890	0.999	0.941	0.812	1,000	0.896
Avarage	0.942	0.937	0.935	0.864	0.829	0.815	0.806	0.723	0.667	0.457	0.641	0.532

Table 9 The classification results of the particulate matter (PM)

	CNN			LSTM			RNN			SVR		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
Normal	1,000	0.551	0.711	1000	0.011	0.021	1000	0.234	0.379	0	0	0
Warning	0.643	0.790	0.709	0.422	0.706	0.529	0.506	0.767	0.610	0.446	0.786	0.569
Critical	0.942	1000	0.970	0.921	1000	0.959	0.937	1,000	0.967	0.941	1000	0.970
Avarage	0.898	0.880	0.875	0.843	0.767	0.710	0.869	0.819	0.795	0.680	0.780	0.721

Table 10 The classification results of the sulfur dioxide (SO_2)

	CNN			LSTM			RNN			SVR		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
Normal	1,000	0.526	0.690	0	0	0	1,000	0.533	0.695	0	0	0
Warning	0.923	0.812	0.864	0.787	0.528	0.632	0.912	0.690	0.786	0.831	0.733	0.779
Critical	0.953	1,000	0.976	0.890	1.000	0.942	0.925	1,000	0.961	0.934	0.999	0.965
Avarage	0.948	0.948	0.945	0.843	0.876	0.852	0.924	0.924	0.918	0.887	0.916	0.900

all algorithms show below 70% classification performance for normal and warning rule types. But CNN slightly outperforms the other algorithms for warning rule types.

We evaluate Table 11, CNN shows above 89% classification performance for warning and critical rules in the CEP engine. It is seen that the normal rule-type outputs are not as good as the others. But CNN outperforms the other algorithms in terms of F-score and accuracy metrics. We can clearly infer that the prediction output of the CNN shows great performance in the CEP engine of the models to detect complex events with given rules. The CNN model obtains more accurate results than other models in complex event detection according to the three rule types considered for all pollutant gases. To compare more clearly the

Table 11 The classification results of the carbon monoxide (CO)

	CNN			LSTM			RNN			SVR		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
Normal	1000	0.687	0.814	1000	0.105	0.191	1000	0.035	0.068	0	0	0
Warning	0.867	0.990	0.924	0.684	0.905	0.779	0.676	0.944	0.788	0.644	0.843	0.730
Critical	0.981	0.987	0.984	0.842	1000	0.914	0.900	0.997	0.946	0.765	1000	0.867
Average	0.928	0.918	0.914	0.800	0.741	0.675	0.810	0.743	0.659	0.523	0.685	0.593

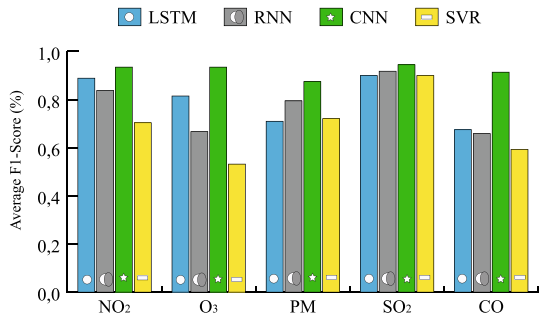

Fig. 3 Average F1-Score of all models

obtained results, we present the average F1-scores of all pollutant gases in Fig. 3.

4.4 Computational complexity evaluation of the CepAIr

The computational complexity of the CepAIr can be summarized by the cost of the prediction unit and CEP engine unit. In the prediction unit, different types of ML are used. In this step, the complexity of the ML changes according to the selected algorithm. The parameters of the model affect the cost of the selected algorithms. Therefore, we assume the worst case complexity which is defined as $O(n^2 + m * n)$ per execution for prediction step [37, 60]. In the complexity of the model, n is defined as the number of input samples, and m is defined as the number of input neurons/unit number of the hidden layer. In the CEP engine, detecting complex events takes $O(n)$ times which can be seen in Algorithm 1. Consequently, the total computational cost of the model can be expressed as $O(n^2 + m * n + n)$.

4.5 End-to-end network delay performance of CepAIr

In this subsection, we evaluate the end-to-end delay performance of the proposed CepAIr on a real-time system based on our previous study [61]. Therefore, we use three widely used network metrics in literature: *i*) the transmission delay, *ii*) the CepAIr processing delay and *iii*) the end-to-end total network delay. At this point, we use the transmission delay times of our previous study to theoretically demonstrate the performance of the CepAIr system on the network. We then measure the processing time of the CepAIr separately according to all models and combine it with the transmission delays in our previous study. Thus, we theoretically obtain the network performance of the proposed system as shown in Table 12. In addition, we give all results according to 95% confidence interval to ensure the statistical soundness of the obtained results.

From Table 12, we see that the average processing delay at each time step t of the proposed CepAIr is measured approximately as 2.8, 4.3, 2.9, and 1.7 ms for the CNN, LSTM, RNN, and SVR models, respectively. At this point, the SVR revealed the least processing delay as it is able to predict without depending on previous data dependencies. When considering the processing delays of CepAIr together with the transmission delays in [61], it can be seen that running CepAIr with LSTM and RNN models caused relatively less network delay. However, when CepAIr is run with CNN and SVR, it reveals higher total network delay compared to LSTM and RNN algorithms. To express in numbers, the total delay times of the network according to the LSTM, RNN, CNN and SVR models are measured as approximately 270, 267, 2,71 and 271 ms, respectively.

Table 12 End-to-end network delay comparison (ms)

	T_{lef}^{up}	T_{lfc}^{up}	T_{lfc}^{down}	T_{lfc}^{down}	T_{e2e}^{trans}	T_{CepAIr}^{proc}	$T_{total}^{network}$
CNN	77.611	119.693	56.100	15.620	269.024 ±2,22	2.840 ±0.05	271.864 ± 14.46
LSTM	75.062	119.541	56.249	15.748	266.600 ±1,20	4.365 ±0.02	270.965 ± 15.39
RNN	73.112	119.641	55.988	16.028	264.769 ±2.03	2.918 ±0.02	267.687 ± 18.66
SVR	76.196	121.949	55.377	16.035	269.557 ±2,75	1.715 ±0.01	271.272 ± 15.08

Based on overall results, the proposed system increased the average network delay by only 1.1%.

5 Conclusion

In this paper, we propose a new fog-based predictive CEP system (CepAIr) that can be a solution to air pollution. The predictive system enables us to inform citizens in urban and rural areas. CepAIr aims to allow decision-makers to take proactive actions by predicting pollutant gas concentrations that cause air pollution. The placement and applicability of CepAIr is illustrated over the three-tiered IoT network model. CepAIr is tested separately in experiments with CNN LSTM, RNN, and SVR models, and the air pollution prediction success and network performance of the CepAIr are separately evaluated. The models takes each pollutant gas and predicts the next pollutant gas level. The experimental results showed that the CNN model has the highest prediction success among the prediction models used in the system. The CEP engine detects the complex events with a boost of prediction unit and sends to alarm label to the decision-makers. On the other hand, when CepAIr's performance on the network is examined in terms of delay, it is seen that it adds only 1.1% additional load to the total network delay. Considering the success of the proposed system in air pollution prediction, the delay rate is considered to be tolerable. As a result, the proposed CepAIr offers a powerful alternative to decision-makers to solve the air pollution problem.

Funding Open access funding provided by the Scientific and Technological Research Council of Türkiye (TÜBİTAK). This work is partially supported by the Scientific and Technological Research Council of Turkey (TUBITAK) under Grant no 118E212.

Data availability The set of data used in the experiments are available at [57].

Declarations

Conflict of interest The authors declare that they have no potential Conflict of interest

Informed consent Not applicable.

References

1. Wang, J., Song, G.: A deep spatial-temporal ensemble model for air quality prediction. Neurocomputing **314**, 198–206 (2018)
2. Li, X., Peng, L., Hu, Y., Shao, J., Chi, T.: Deep learning architecture for air quality predictions. Environ. Sci. Pollut. Res. **23**(22), 22 408-22 417 (2016)
3. EPA, Criteria air pollutants, https://www.epa.gov/criteria-air-pollutants, Accessed: 2021-03-10
4. Athira, V., Geetha, P., Vinayakumar, R., Soman, K.: Deepairnet: applying recurrent networks for air quality prediction. Proc. Comput. Sci. **132**, 1394–1403 (2018)
5. EPA, "Air pollution," https://www.who.int/news-room/air-pollution, Accessed: 2021 02 15
6. Bellinger, C., Mohomed Jabbar, M.S., Zaïane, O., Osornio-Vargas, A.: A systematic review of data mining and machine learning for air pollution epidemiology. BMC Public Health **17**, 1–19 (2017)
7. Delavar, M.R., Gholami, A., Shiran, G.R., Rashidi, Y., Nakhaeizadeh, G.R., Fedra, K., Hatefi Afshar, S.: A novel method for improving air pollution prediction based on machine learning approaches: a case study applied to the capital city of tehran. ISPRS Int. J. Geo-Inf. **8**(2), 99 (2019)
8. Wen, C., Liu, S., Yao, X., Peng, L., Li, X., Hu, Y., Chi, T.: A novel spatiotemporal convolutional long short-term neural network for air pollution prediction. Sci. Total Environ. **654**, 1091–1099 (2019)
9. Kang, G.K., Gao, J.Z., Chiao, S., Lu, S., Xie, G.: Air quality prediction: big data and machine learning approaches. Int. J. Environ. Sci. Dev **9**(1), 8–16 (2018)
10. Liu, H., Yin, S., Chen, C., Duan, Z.: Data multi-scale decomposition strategies for air pollution forecasting: a comprehensive review. J. Clean. Prod. **277**, 124023 (2020)
11. Flouris, I., Giatrakos, N., Deligiannakis, A., Garofalakis, M., Kamp, M., Mock, M.: Issues in complex event processing: status and prospects in the big data era. J. Syst. Softw. **127**, 217–236 (2017)
12. Akbar, A., Khan, A., Carrez, F., Moessner, K.: Predictive analytics for complex iot data streams. IEEE Internet Things J. **4**(5), 1571–1582 (2017)
13. Esmaeilyfard, R., Naderi, M.: Distributed composition of complex event services in iot network. J. Supercomput. (2020). https://doi.org/10.1007/s11227-020-03498-2
14. Okay, F.Y., Kok, I., Guzel, M., Ozdemir, S.: "Fog computing-based complex event processing for internet of things," in Big Data-Enabled Internet of Things, ser. Computing, A. Y. Z. Muhammad Usman Shahid Khan, Samee U. Khan 2, Ed. Institution of Engineering and Technology, (2019), pp. 137–173. [Online]. Available: https://digital-library.theiet.org/content/books/10.1049/pbpc025e_ch8
15. Terroso-Saenz, F., Valdes-Vela, M., Sotomayor-Martinez, C., Toledo-Moreo, R., Gomez-Skarmeta, A.F.: A cooperative approach to traffic congestion detection with complex event processing and vanet. IEEE Trans. Intell. Trans. Syst. **13**(2), 914–929 (2012)
16. Akbar, A., Carrez, F., Moessner, K., Zoha, A.: Predicting complex events for pro-active iot applications, in,: IEEE 2nd World

Forum on Internet of Things (WF-IoT). IEEE **2015**, 327–332 (2015)
17. Bruns, R., Dunkel, J., Masbruch, H., Stipkovic, S.: Intelligent m2m: complex event processing for machine-to-machine communication. Exp. Syst. Appl. **42**(3), 1235–1246 (2015)
18. Kök, I., Şimşek, M.U., Özdemir, S.: A deep learning model for air quality prediction in smart cities, in. IEEE International Conference on Big Data (Big Data) **2017**, 1983–1990 (2017)
19. Fülöp, L.J., Beszédes, Á., Tóth, G., Demeter, H., Vidács, L., Farkas, L.: "Predictive complex event processing: a conceptual framework for combining complex event processing and predictive analytics," in Proceedings of the Fifth Balkan Conference in Informatics, (2012), 26–31
20. Schwegmann, B., Matzner, M., Janiesch, C.: "A method and tool for predictive event-driven process analytics." in Wirtschaftsinformatik, (2013), 46
21. Cabanillas, C., Di Ciccio, C., Mendling, J., Baumgrass, A.: "Predictive task monitoring for business processes," in International Conference on Business Process Management. Springer, (2014), 424–432
22. Wang, Y., Cao, K.: A proactive complex event processing method for large-scale transportation internet of things. Int. J. Distrib. Sensor Netw. **10**(3), 159052 (2014)
23. Wang, Y., Gao, H., Chen, G.: Predictive complex event processing based on evolving bayesian networks. Pattern Recognit. Lett. **105**, 207–216 (2018)
24. Nechifor, S., Târnaucă, B., Sasu, L., Puiu, D., Petrescu, A., Teutsch, J., Waterfeld, W., Moldoveanu, F.: "Autonomic monitoring approach based on cep and ml for logistic of sensitive goods," in IEEE 18th International Conference on Intelligent Engineering Systems INES 2014. IEEE, (2014), 67–72
25. Christ, M., Krumeich, J., Kempa-Liehr, A.W., Integrating predictive analytics into complex event processing by using conditional density estimations, in,: IEEE 20th International Enterprise Distributed Object Computing Workshop (EDOCW). IEEE **2016**, 1–8 (2016)
26. Emerson, R.J., Hossen, J., Ervina, E., Tawsif, K., Jesmeen, M.: Broadband network fault prediction using complex event processing and predictive analytics techniques. J. Eng. Sci. Technol. **15**(4), 2289–2300 (2020)
27. Xing, T., Vilamala, M. R., Garcia, L., Cerutti, F., Kaplan, L., Preece, A., Srivastava, M.: "Deepcep: Deep complex event processing using distributed multimodal information," in 2019 IEEE International Conference on Smart Computing (SMARTCOMP). IEEE, 2019, 87–92
28. Yadav, P., Sarkar, D., Salwala, D., Curry, E.: "Traffic prediction framework for openstreetmap using deep learning based complex event processing and open traffic cameras," arXiv preprint arXiv:2008.00928, (2020)
29. Díaz, G., Macià, H., Valero, V., Boubeta-Puig, J., Cuartero, F.: An intelligent transportation system to control air pollution and road traffic in cities integrating cep and colored petri nets. Neural Comput. Appl. **32**(2), 405–426 (2020)
30. Boubeta-Puig, J., Díaz, G., Macià, H., Valero, V., Ortiz, G.: Medit4cep-cpn: an approach for complex event processing modeling by prioritized colored petri nets. Inf. Syst. **81**, 267–289 (2019)
31. Semlali, B.-E.B., El Amrani, C., Ortiz, G., Boubeta-Puig, J., Garcia-de Prado, A.: Sat-cep-monitor: an air quality monitoring software architecture combining complex event processing with satellite remote sensing. Comput. Electrical Eng. **93**, 107257 (2021)
32. Brazalez, E., Macia, H., Diaz, G., Baezaromero, M., Valero, E., Valero, V.: Fume: an air quality decision support system for cities based on cep technology and fuzzy logic. Appl. Soft Comput. **129**, 109536 (2022)
33. Macià, H., Díaz, G., Boubeta-Puig, J., Valero, E., Valero, V.: Combining fuzzy logic and cep technology to improve air quality in cities. in International Conference on Computational Science. Springer, 559–565 (2019)
34. Liu, Y., Yu, W., Gao, C., Chen, M.: An auto-extraction framework for cep rules based on the two-layer lstm attention mechanism: a case study on city air pollution forecasting. Energies **15**(16), 5892 (2022)
35. Yemson, R., Kabir, S., Thakker, D., Konur, S.: Ontology development for detecting complex events in stream processing: use case of air quality monitoring. Computers **12**(11), 238 (2023)
36. Zhang, J., Man, K.: "Time series prediction using rnn in multi-dimension embedding phase space," in SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 98CH36218), **2**. IEEE, 1868–1873 (1998)
37. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
38. Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.C.: Convolutional lstm network: a machine learning approach for precipitation nowcasting. Adv. Neural Inf. Proc. Syst. **2015**, 802–810 (2015)
39. Zhu, X., Sobihani, P., Guo, H.: "Long short-term memory over recursive structures," in International Conference on Machine Learning. PMLR, 1604–1612 (2015)
40. Guzel, M., Kok, I., Akay, D., Ozdemir, S.: Anfis and deep learning based missing sensor data prediction in iot. Concurr. Comput.: Practice Exp. **32**(2), e5400 (2020)
41. Qin, D., Yu, J., Zou, G., Yong, R., Zhao, Q., Zhang, B.: "A novel combined prediction scheme based on cnn and lstm for urban pm 2.5 concentration," IEEE Access, **7**, 20 050–20 059, (2019)
42. Canizo, M., Triguero, I., Conde, A., Onieva, E.: Multi-head cnn-rnn for multi-time series anomaly detection: An industrial case study. Neurocomputing **363**, 246–260 (2019)
43. Pan, H., He, X., Tang, S., Meng, F.: An improved bearing fault diagnosis method using one-dimensional cnn and lstm. J. Mech. Eng. **64**(7–8), 443–452 (2018)
44. Masci, J., Meier, U., Ciresan, D., Schmidhuber, J., Fricout, G.: "Steel defect classification with max-pooling convolutional neural networks," in The 2012 International Joint Conference on Neural Networks (IJCNN). IEEE, (2012), 1–6
45. Li, T., Zhang, Z., Chen, H.: Predicting the combustion state of rotary kilns using a convolutional recurrent neural network. J. Process Control **84**, 207–214 (2019)
46. Ho, C.-H., Lin, C.-J.: Large-scale linear support vector regression. J. Mach. Learn. Res. **13**(1), 3323–3348 (2012)
47. Vapnik, V.: The Nature of Statistical Learning Theory. Springer science & business media, Berlin (2013)
48. Kakarash, Z.A., Ezat, H.S., Omar, S., Ahmed, N.F.: Time series forecasting based on support vector machine using particle swarm optimization. Int. J. Comput. **21**(1), 76–88 (2022)
49. Lee, S., Kim, C.K., Kim, D.: Monitoring volatility change for time series based on support vector regression. Entropy **22**(11), 1312 (2020)
50. AirNow, "Air quality index (aqi) basics," https://www.airnow.gov/aqi/aqi-basics/, accessed: 2021-03-10
51. Espertech, Link, 2021 (accessed March 20, 2021), https://www.espertech.com/esper/
52. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M. et al.: "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," arXiv preprint arXiv:1603.04467, (2016)
53. Chollet, F., et al.: "Keras: Deep learning library for theano and tensorflow," URL: https://keras. io/k, **7**,(8),T1, (2015)
54. Garreta, R., Moncecchi, G.: Learning Scikit-Learn: Machine Learning in Python. Packt Publishing Ltd, Birmingham (2013)

55. Mathew, A.: Benchmarking of complex event processing engine-esper, Technical Report IITB/CSE/2014/April/61, Department of Computer Science and Engineering. Tech. Rep, Indian Institute of Technology Bombay (2014)
56. Consortium, T.: "Citypulse annual report," The CityPulse Consortium, (2016)
57. I. C. P. Dataset, Download link, 2020 (accessed December 3, 2020), http://iot.ee.surrey.ac.uk:8080/
58. U. S. E. P. A. O. of Air Quality Planning, Standards, U. S. E. P. A. Monitoring, D. A. Division, U. S. E. P. A. O. of Air Quality Planning, S. T. S. Division, and U. S. E. P. A. A. Q. T. A. Group, National air quality and emissions trends report. US Environmental Protection Agency, Office of Air and Radiation, Office of ..., (2003)
59. Chai, T., Draxler, R.R.: Root mean square error (rmse) or mean absolute error (mae)?-arguments against avoiding rmse in the literature. Geosci. Model Dev. **7**(3), 1247–1250 (2014)
60. Sak, H., Senior, A., Beaufays, F.: "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," arXiv preprint arXiv:1402.1128, (2014)
61. Kök, I., Özdemir, S.: Deepmdp: a novel deep-learning-based missing data prediction protocol for iot. IEEE Internet Things J. **8**(1), 232–243 (2021)

Mehmet Ulvi Şimsek received his MSc and PhD degrees in Computer Science from Gazi University in 2012 and 2021, respectively. He is currently working as a Senior Lead Engineer in defence industry. His current research interests include the Internet of Things (IoT), Big Data, Deep Learning, AI-enabled IoT, Data Science, and Data Analytics.

İbrahim Kök received his MSc and PhD degrees in Computer Science from Gazi University in 2015 and 2020, respectively. He is currently an Assistant Professor at the Department of Computer Engineering, Pamukkale University, Denizli. His current research interests include the Internet of Things (IoT), Deep Learning, AI-enabled IoT, Explainable AI, and Data Analytics.

Suat Özdemir is with the Department of Computer Engineering at Hacettepe University, Ankara, Turkey. He received his MSc degree in Computer Science from Syracuse University (August 2001) and PhD degree in Computer Science from Arizona State University (December 2006). His current research interests include Internet of Things, Data Analytics, Artificial Intelligence, and Network Security.