Toward Formal Methods for Smart Cities

Meiyi Ma, Vanderbilt University
John A. Stankovic and Lu Feng, University of Virginia

How can the advantages of formal methods be brought to emerging smart cities? We discuss several core challenges and our recent efforts as the first step toward developing novel formal methods to ensure safety and performance in smart cities.

The prevalence of the Internet of Things and cyberphysical systems (CPSs) has enabled the emergence of smart cities around the world, where a vast amount of sensing data and smart services are utilized to improve citizens' safety, wellness, and quality of life.[1,2] Various smart city operation control centers (for example, Microsoft's CityNext, IBM's Rio de Janeiro Operations Center, and Cisco's Smart+Connected Operations Center) have been developed to support decision making in smart cities based on real-time sensing data about city states (such as traffic and air pollution).

While significant research efforts have been spent toward building smarter services, sensors, and infrastructures in cities, the research challenge of how to ensure that a city's real-time operations satisfy safety and performance requirements has received only scant attention. Failure to check such requirements can lead to conflicts among smart services or even catastrophic consequences.[3–5] This article discusses several core challenges in developing novel formal methods for ensuring safety and performance in smart cities. Specifically, we focus on addressing three key research questions.

Digital Object Identifier 10.1109/MC.2021.3082991
Date of current version: 27 August 2021

First, how should we monitor whether city states satisfy a wide range of city requirements at runtime? If a requirement violation is detected by the monitor, the city operators and smart service providers can take actions to change the states, such as improving traffic performance, rejecting unsafe actions, sending alarms to police, and so on. The key challenges of developing such a monitor include how to use an expressive formal language to specify smart city requirements so that they can be understood by machines and developing ways to efficiently monitor requirements that may involve multiple sensor data streams (for example, some requirements are concerned with thousands of sensors in a smart city).

Second, how can we predict a city's future states and check if the prediction satisfies city requirements? With this capability, city operators may take actions in advance to prevent such predicted future requirement violations. A key challenge of predictive monitoring is how to account for the inherent uncertainty (for example, due to sensor and environmental noise, unexpected events, accidents, and human behaviors) in smart cities.

AS DEEP LEARNING TECHNIQUES ARE INCREASINGLY USED IN SMART CITY APPLICATIONS, HOW DO WE GUARANTEE THAT THE RESULTS WILL SATISFY CITY REQUIREMENTS?

Third, as deep learning techniques are increasingly used in smart city applications, how do we guarantee that the results will satisfy city requirements? For example, recurrent neural networks (RNNs) have made great achievements for sequential prediction tasks in cities [for example, forecasting the air quality index (AQI)]. Can we enforce that the learned sequence predictions must satisfy certain desired properties in smart cities? In the following sections, we elaborate on these research questions and present our solutions and insights to help lay a foundation for ensuring safety and performance in smart cities.

RUNTIME MONITORING OF SPATIAL–TEMPORAL CITY REQUIREMENTS

We collected and analyzed more than 1,000 real-world city requirements from multiple cities (for example, extracted from city regulations, standards, codes, and laws) in different domains, including transportation, energy, environment, emergency, and public safety. Table 1 shows some example requirements. We found that most city requirements highlight spatial [for example, the distance from points of interest (PoIs)] and temporal constraints (such as real-time deadlines): for example, "The average noise level within 1 mi of schools should be fewer than 50 dB."

Existing formal specification languages, such as signal temporal logic (STL) and its extensions [such as signal spatiotemporal logic (SSTL), spatial-temporal logic, and spatial temporal reach and escape logic (STREL)][6] can be used only to express a subset of city requirements. However, they are not expressive enough to specify the aggregation requirements (such as "the average noise level") and counting (for example, "on 90% of the roads") of signals in the spatial domain, which are commonly used in city requirements.

To address this limitation, we proposed a novel spatial aggregation STL (SaSTL),[7] which extends STL with logical operators for spatial aggregation and counting. SaSTL can be used to specify the PoIs, physical distance, spatial relations of the PoIs and sensors, aggregation of the signals over locations, degree/percentage of satisfaction, and temporal elements in a very flexible spatial–temporal scale. The results of comparing the coverage of different formal specification languages for expressing 1,000 real-world city requirements show that SaSTL has a much higher coverage expressiveness (95%) than STL (18.4%), SSTL (43.1%), or STREL (43.1%).

We developed a framework for the runtime monitoring of smart city requirements expressed in SaSTL. Figure 1 shows an overview of the framework. We envision that such a framework would operate in a smart city's central control center where sensor data about city states across various locations are available in real time. The framework can monitor different city data streams (such as noise level and traffic volume) over the spatial and temporal domains at the runtime and check them against a set of city requirements formalized in SaSTL.

Such runtime monitoring results can then be used to support smart cities' decision making. The framework is based on our novel and efficient monitoring algorithms for SaSTL. In particular, we developed two methods to speed up the monitoring performance: 1) dynamically prioritizing the monitoring based on the cost functions assigned to the nodes of the syntax tree and 2) parallelizing the monitoring of spatial operators among multiple locations and/or sensors.

Based on our SaSTL monitoring framework, we implemented a user-friendly tool to support the decision making of different stakeholders in smart cities. The tool allows users (for example, city decision makers or citizens) without any formal method background to specify city requirements and monitor city performance easily. Figure 2 shows the tool's user interface and the four steps of using the tool:

WE DEVELOPED A FRAMEWORK FOR THE RUNTIME MONITORING OF SMART CITY REQUIREMENTS EXPRESSED IN SASTL.

1. selecting the monitoring area and PoIs (in the blue box)
2. setting up the city data sources
3. specifying the city requirements with structured language, which are automatically

TABLE 1. An example of city requirements from different domains.[7]

Domain	Example
Transportation	There is a limit for vehicle idling to 1 min adjacent to any school, pre-K to 12th grade, public or private, in the City of New York.
	The engine, power, and exhaust mechanism of each motor vehicle shall be equipped, adjusted, and operated to prevent the escape of a trail of visible fumes or smoke for more than 10 consecutive seconds.
	Sightseeing buses are prohibited from using all bus lanes between the hours of 7:00 and 10:00 a.m. on weekdays.
Energy	The system is operated to maintain a zone temperature down to 55 °F or up to 85 °F.
	The total leakage shall be less than or equal to 4 ft^3/ min / 100 ft^2 of conditioned floor area.
Environment	LA Sec. 111.03 minimum ambient noise level table is used: zones M2 and M3 — day: 65 dB(A); night: 65 dB(A).
	The total amount of HCHO emissions should be less than 0.1 mg/m^3 within an hour, and the total amount of PM10 emissions should be less than 0.15 mg/m^3 within 24 hours.
Emergency	New York City authorized emergency vehicles may disregard four primary rules regarding traffic.
	At least one ambulance should be equipped per 30,000 population (counted by area) to obtain the shortest radius and fastest response time.
Public safety	Security staff shall visit at least once per week in public schools.

Key elements: temporal, spatial, aggregation, entity, condition, and comparison.

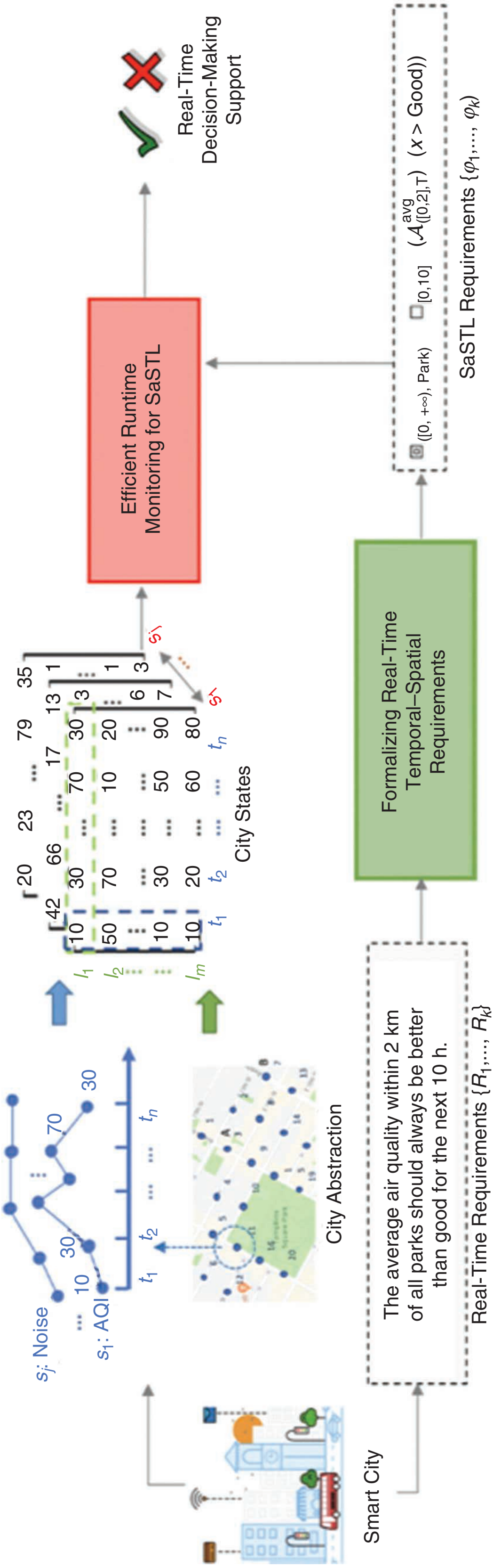


FIGURE 1. The runtime monitoring of the smart city requirements specified in SaSTL. (Source: Ma et al.[7])

translated into formal SaSTL properties
4. runtime monitoring using SaSTL algorithms, with the results displayed.

We evaluated the SaSTL monitoring tool using three real smart city scenarios (New York and Chicago from the United States as well as Aarhus from Denmark) with large-scale real sensing data (for example, up to 10,000 sensors used in one requirement). The results show that the SaSTL monitor has the potential to help identify safety violations and support city managers and citizens in making decisions. In our simulated experiments, the SaSTL monitor can help improve the city's performance with a significant reduction of computation time compared with previous approaches.

We envision this tool can be used by different stakeholders in smart cities, including but not limited to the following:

- *City managers and decision makers*: In the city operating center, with city data collected in real time, the tool is able to help city managers and decision makers to monitor the data at runtime. It also helps the city center detect conflicts and provides support for decision makers by showing the tradeoffs of satisfaction degrees among potential solutions.
- *City planners*: City planners, either from the government to make long-term policies or from a company to make a short-term event plan, are able to use the tool to verify the past city data with their requirements and make preparations to prevent violations.

- *Service designers*: Smart services are designed by different stakeholders, including the government, companies, and private parties, and often they are not aware of all of the other smart services. However, with the monitor, they can test the influence of their services and adjust them to better serve the city.
- *Everyday citizens*: The tool can also provide a service to everyday citizens. People without any technical background are able to specify their own requirements and check them with the data to find out in which areas of the city and period of the day their requirements are satisfied so they can make daily plans. For example, a citizen can specify an environmental requirement with his/her preferred AQI and traffic conditions, check the city data with the requirements, and make up traveling agenda accordingly.

We are currently working with project partners to deploy the tool in the City of Newark, New Jersey, to demonstrate its impact via real-world applications.

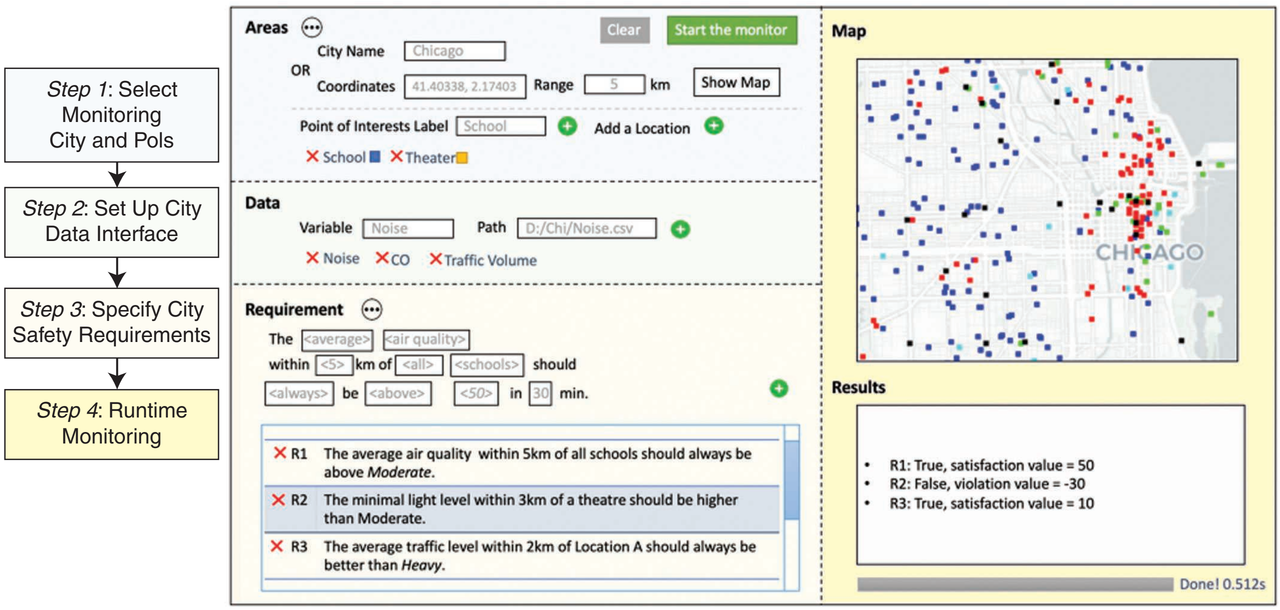
PREDICTIVE MONITORING FOR SMART CITIES

Deep learning techniques have been increasingly applied to predict smart city states (for example, air quality forecasting). However, previous works mostly focus only on generating predictions and rarely account for the uncertainty inherent in smart cities (such as sensing and environmental noise, unexpected events, and accidents).

We tackle this challenge by developing an STL with uncertainty (STL-U)-based predictive monitoring approach[8] for CPSs, including smart cities. The predictive monitoring framework interacts with a smart city control center to continuously predict future city states and monitor if predictions satisfy city requirements. If it forecasts a potential city requirement violation in a future state, it would support the decision system in a control center to choose actions (for example, issuing alarms or controlling traffic signals) to prevent such a requirement violation. Specifically, our predictive monitoring approach advances the state of the art from the following two aspects: monitoring and prediction.

Monitoring

STL and its extensions have been applied for monitoring smart city requirements. However, existing methods mostly focus on monitoring a single multivariable signal and cannot be directly applied



(a) (b)

FIGURE 2. The (a) steps and (b) user interface of the SaSTL monitoring tool for smart cities. (Source: Ma et al.[7])

for monitoring the Bayesian sequential predictions. To address this challenge, we formalized the notion of a flowpipe signal to characterize the prediction outputs of Bayesian deep learning and developed a new logic, named *STL-U*, for reasoning about the correctness of flowpipe signals.

STL-U can be used to specify city requirements with uncertainty, such as "With a 90% confidence level, the predicted AQI in the next 10 h should always be below 100." We also developed algorithms for computing the confidence level that guarantees an STL-U property is satisfied by the given flowpipe signals. Such results can provide smart city decision makers with meaningful confidence guarantees about the predictions of city future states satisfying the city requirements.

Prediction

Various machine learning and statistical analysis techniques (for example, neural networks and autoregressive integrated moving average) have been popularly applied to predict the future states of CPSs across different application domains. RNN-based sequential prediction has been popularly applied to smart cities. However, existing results mostly use deterministic RNNs, which generate a single sequence of predictions and do not capture the uncertainty in smart cities.

Recent advances, such as Bayesian deep learning techniques, can adapt the prediction output stochastically as a sequence of posterior probability distributions over a finite discrete time domain. However, existing methods often use the loss functions of deep learning models (such as the mean square error, negative log likelihood, and Kullback–Leibler divergence) as the only metrics for the uncertainty estimation, which tend to overestimate or underestimate the uncertainty level. Furthermore, these metrics treat the uncertainty estimation of each individual value in a predicted sequence separately and, thus, lack an integrated view about the uncertainty of sequential predictions.

To address this challenge, we developed novel logic-based criteria to measure uncertainty that are sufficiently general to be applied to any sequential prediction models. Our approach uses these logic-calibrated uncertainty measurements to select and tune the uncertainty estimation schema in deep learning models.

Figure 3 shows an overview of the STL-U-based predictive monitoring approach. It first takes a city's historical states (for example, the AQI in the past 5 h) as inputs and returns the city's future states (such as the predicted AQI in next 2 h) via an RNN-based Bayesian sequential prediction model. The predicted future states are represented by a sequence of distributions. At each predicted time point, it shows a range of the potential values under a given confidence level. Then, the STL-U monitor takes the predicted states and formalized city requirements as inputs and returns the projected verification results over the future time interval.

At training time (the flow marked by the orange dashed lines in Figure 3), our predictive monitoring approach conducts model selection and tuning using STL-U criteria to obtain a well-calibrated uncertainty estimation schema for the RNN-based Bayesian sequential prediction. Intuitively, the satisfaction degree of the predicted sequence (that is, the predicted future states) should be same as the satisfaction degree of the target sequence (that is, the ground truth values). STL-U criteria are designed to measure the loss based on the monitoring results and, thus, evaluate the quality of the uncertainty estimation

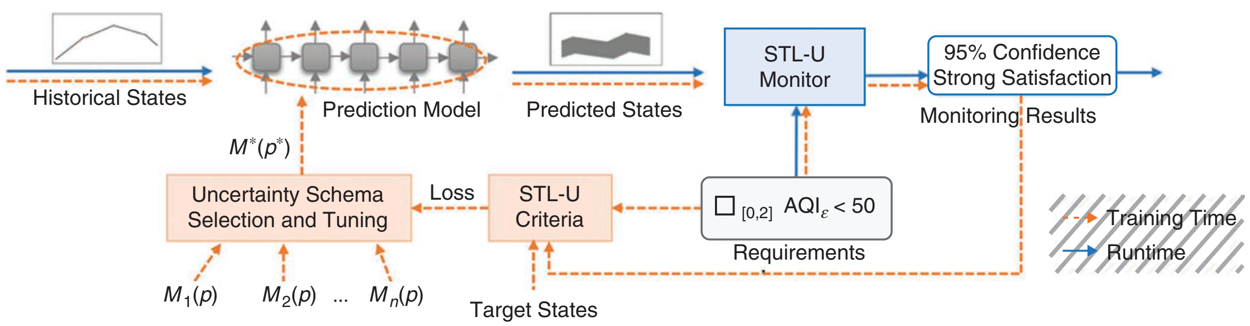


FIGURE 3. The predictive monitoring for smart cities. (Source: Ma et al.[8])

schema. In this way, the uncertainty estimation schema with the smallest STL-U loss is selected.

At runtime (the flow marked by the blue lines in Figure 3), our approach outputs the current and future monitoring results to support the smart city control center. As a real-time operational scenario, our approach runs as a continuous iterative process. For example, for the predictive monitoring of the AQI in a smart city, at time t, our approach first predicts the AQI for the future 3 h from time t and monitors if the predictions satisfy the city requirements; after a certain period d (for example, 30 min), our approach predicts the AQI for the future 3 h from $t + d$ and checks if the new predictions satisfy the requirements. In this way, the STL-U-based predictive monitoring framework provides the continuous predictive monitoring of city states for smart city decision makers.

We evaluated the performance of our approach using real city data sets and simulations. The results show that our approach significantly improves the simulated city's safety and performance, and the use of STL-U logic-based criteria leads to improved uncertainty calibration in various Bayesian deep learning models. For example, Figure 4 compares F1 scores on the accuracy of the requirements verification (that is, if the predicted flowpipe satisfies/violates the requirement when the target sequence satisfies/violates the requirement) using three RNNs trained by different loss functions. The results show that all STL-U criteria ($\mathcal{L}_{sat}$ and $\mathcal{L}_{cf}$) outperform the accuracy-based criterion ($\mathcal{L}_{acc}$ and $\mathcal{L}_{ht}$) significantly.

The STL-U predictive monitoring approach demonstrates the feasibility of integrating formal methods and Bayesian deep learning for the predictive monitoring of safety and performance requirements in smart cities. In addition, the proposed STL-U criteria can be applied for the uncertainty estimation in a wide range of deep learning applications. Compared with traditional uncertainty estimation methods,[9] the proposed logic-based solution can lead to better uncertainty calibration for sequential prediction tasks.

FORMAL LOGIC-ENFORCED DEEP LEARNING FOR SMART CITIES

RNNs have made great achievements for sequential prediction tasks. In practice, the target sequence values often follow certain model properties or patterns (for example, reasonable ranges for a variable, how consecutive changes in variables are realistic, how resource constraints limit values for variables, temporal correlations among multiple variables, the existence of an event within a certain time, unusual cases with no or a very limited amount of data available in the training set, and so on). However, RNNs cannot guarantee that their learned distributions satisfy these properties.

It is even more challenging for the prediction of large-scale and complex CPSs, such as smart cities. Failure to produce outcomes that meet these properties will result in inaccurate and even meaningless results. To address this challenge, we developed a novel formal logic-enforced deep learning framework, named *STL-enforced*

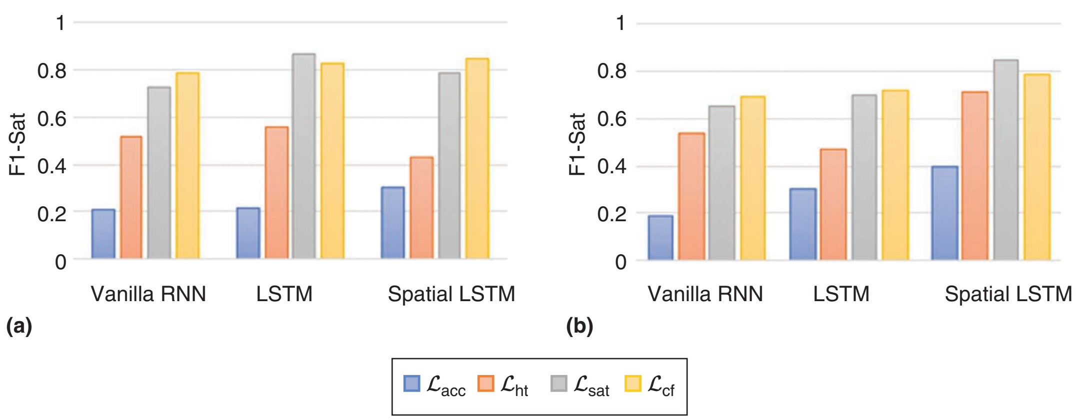


FIGURE 4. A comparison of F1 scores on the consistence of verification between predicted flowpipes and target sequences using different RNN-based prediction models with different loss functions for (a) air quality and (b) traffic volume. acc: accuracy; sat: satisfaction; ht: heteroscedastic; cf: confidence. (Source: Ma et al.[8]; used with permission.)

multivariate RNN (*STLnet*).[10] It guides the RNN learning process with auxiliary knowledge of model properties and produces a more robust model for improved future predictions.

Figure 5 shows an overview of the STLnet framework, which is built with a teacher and student network. The teacher network is equipped with an STL trace generator, which incorporates the formalized model properties into the learning process. The main idea is that whenever the student network fails to predict a trace (sequence) that follows the model properties, the teacher network generates a trace that is close to the trace returned by the student network and satisfies the model properties simultaneously. The student network then updates its parameters by learning from both the target trace and outcome of the teacher network.

In the training phase, the goal is to teach STLnet to learn from the "correct" traces, which includes three major steps:

- *Step 1*: The student network construction starts with the basic student network, that is, a general multivariate RNN.
- *Step 2*: The teacher network construction generates a trace that satisfies the model properties expressed in STL and has the shortest distance to the original prediction. Table 2 shows some example model properties for smart city applications.
- *Step 3*: Back propagation with a loss function is designed with two parts to guide the student network to balance between emulating the teacher's output and predicting the target trace.

The network is trained iteratively by repeating Steps 2 and 3 until convergence.

In the testing phase, we can use either the distilled student or teacher network after a final projection. Our results show that both models substantially improve over the base network that is trained without STL-specified properties. In practice, the teacher network can guarantee the satisfaction of model properties, while the student network is more lightweight and efficient.

We evaluated the performance of STLnet using large-scale, real-world city data that include 1.3 million instances of six pollutants (that is, PM2.5, PM10, CO, SO_2, NO_2, and O_3) collected from 130 locations in Beijing every hour between 1 May 2014 and 30 April 2015. To build the LSTM network, we regard one pollutant from one location as one variable and concatenate all variables from the same time unit.

Next, we specify important model properties, including reasonable ranges, consecutive changes, correlations among different pollutants and locations, and so on. Figure 6 shows the comparison results (with respect to the root-mean-square error and satisfaction

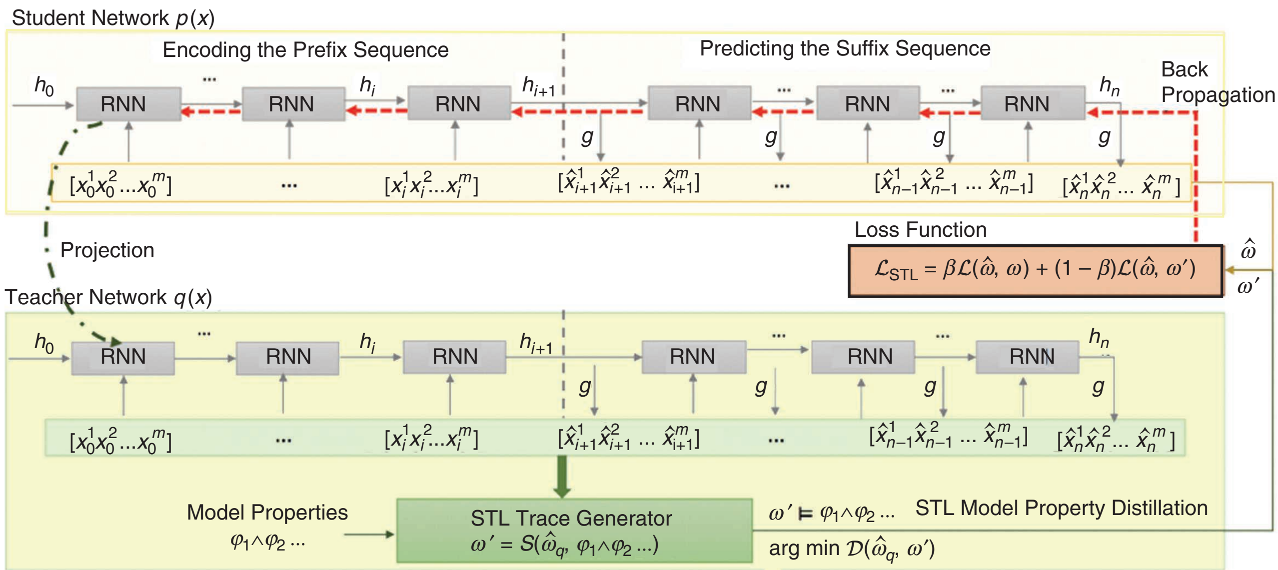


FIGURE 5. The STLnet. (Source: Ma et al.[10])

rate of model properties), which indicate that STLnet improves the accuracy and robustness of RNNs in a real-world CPS application, especially in cases of noisy/missing sensing data, and long-term prediction.

The proposed STLnet is broadly applicable to various sequential prediction tasks beyond smart cities. This work shows the promise of leveraging formal methods to enhance the robustness and reliability of deep learning.

While tremendous progress has been made in advancing formal methods for CPSs, the research area of formal methods for smart cities is still in its infancy. In this article, we presented our recent efforts as the first step in developing novel formal methods to guarantee safety and performance in smart cities. There are many open research problems in this exciting new area that need further study:

- improving the scalability of formal methods for the runtime

TABLE 2. Examples of model properties and their corresponding logic formulas.[10]

Property type	Example	STL formula
Reasonable range	The traffic volume on a road can never exceed the road capacity.	$\square_{[0,24]}(x_1 < \alpha_1) \wedge \cdots \wedge \square_{[0,24]}(x_n < \alpha_n)$
Consecutive changes	The number of people in a shopping mall should not increase or decrease by more than 1,000 in 10 min if the number of exits fewer than 5.	$y < 5 \rightarrow \square_{[0,10]}(\Delta x < 1{,}000)$
Resource constraint	The total energy distributed to all buildings should be less than *e*.	$\square_{[0,24]}\,\mathrm{sum}\,(x_1,\ldots x_n) < e$
Variable and temporal correlation	For two consecutive intersections on a one-way-direction road, if there are 10 cars passing intersection A, then there should be at least 10 cars passing intersection B within the next 5 min.	$(x_1 > 10 \rightarrow \Diamond_{[0,5]}(x_2 > 10)) \wedge \cdots \wedge (x_n > 10 \rightarrow \Diamond_{[0,5]}(x_{n+1} > 10))$
Existence	They should be at least one patrol car around a school every day.	$\Diamond_{[0,24]}\, x_1 \geq 1 \wedge \cdots \wedge \Diamond_{[0,24]}\, x_n \geq 1$
Unusual cases	If there is a concert on Friday, the number of people in the nearby shopping mall will increase by at least 200 within 2 h.	$x_{\mathrm{Event}} = \mathrm{True} \wedge x_{\mathrm{Day}} = \mathrm{Fri} \rightarrow \Diamond_{[0,2]}\, \Delta x > 200.$

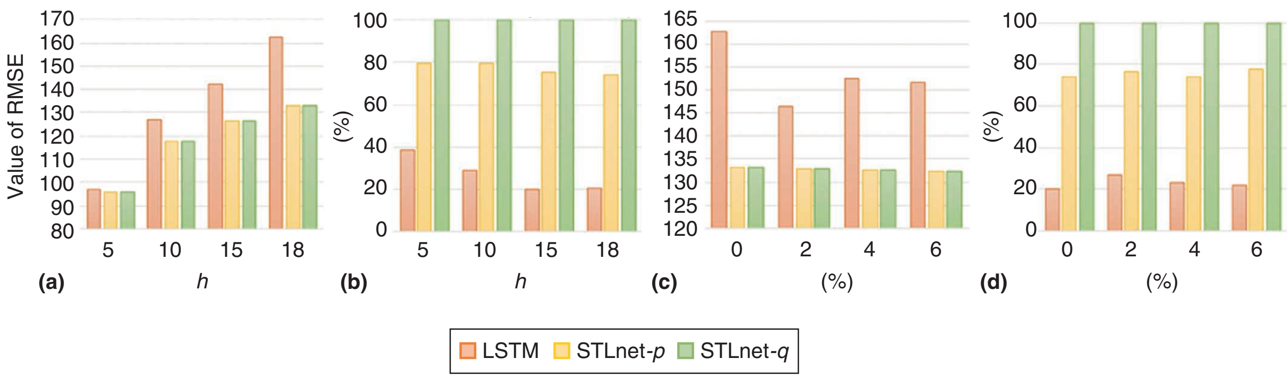


FIGURE 6. A comparison of the root-mean-square error (RMSE) and satisfaction rate among the LSTM, STLnet-*p* (the student network), and STLnet-*q* (the teacher network): the prediction lengths of the (a) RMSE and (b) satisfaction rate as well as the missing data percentages of the (c) RMSE and (d) satisfaction rate. (Source: Ma et al.[10]; used with permission.)

ABOUT THE AUTHORS

MEIYI MA is an assistant professor of computer science at Vanderbilt University, Nashville, Tennessee, 37235, USA. Her research interests include cyberphysical systems, deep learning, and formal methods. Ma received a Ph.D. in computer science from the University of Virginia. This work was done when she was at the University of Virginia. Contact her at meiyi@virginia.edu.

JOHN A. STANKOVIC is the BP America Professor in the Computer Science Department at the University of Virginia, Charlottesville, Virginia, 22903, USA, and director of the Link Lab. His research interests include smart and connected health, cyberphysical systems, and the Internet of Things. Stankovic received a Ph.D. from Brown University. He is a Fellow of IEEE and the Association for Computing Machinery. Contact him at stankovic@virginia.edu.

LU FENG is an assistant professor of computer science at the University of Virginia, Charlottesville, Virginia, 22903, USA. Her research interests include cyberphysical systems and formal methods. Feng received a Ph.D. in computer science from the University of Oxford. Contact her at lu.feng@virginia.edu.

monitoring of smart city states, which involve large-scale sensing data from hundreds of thousands of geographically sparsely distributed sensors
- making the use of tools and solutions easier for city stakeholders without a background knowledge of formal methods
- applying formal methods (for example, model-based development) to support the development of smart services and integration in smart cities
- leveraging formal methods (such as robustness certification) to create reliable deep learning models for smart cities
- developing formal methods to measure and validate social-aware fairness, accountability, transparency, and tradeoffs in smart cities.

ACKNOWLEDGMENTS

This work was supported in part by National Science Foundation grants CCF-1942836 and CNS-1952096.

REFERENCES

1. M. Batty et al., "Smart cities of the future," *Eur. Phys. J. Special Topics*, vol. 214, no. 1, pp. 481–518, 2012. doi: 10.1140/epjst/e2012-01703-3.
2. M. Ma, S. M. Preum, M. Y. Ahmed, W. Tärneberg, A. Hendawi, and J. A. Stankovic, "Data sets, modeling, and decision making in smart cities: A survey," *ACM Trans. Cyber-Phys. Syst.*, vol. 4, no. 2, pp. 1–28, 2019. doi: 10.1145/3355283.
3. M. Ma, S. Masud Preum, W. Tarneberg, M. Ahmed, M. Ruiters, and J. Stankovic, "Detection of runtime conflicts among services in smart cities," in *Proc. 2016 IEEE Int. Conf. Smart Comput. (SMARTCOMP)*, pp. 1–10. doi: 10.1109/SMARTCOMP.2016.7501688.
4. M. Ma, S. M. Preum, and J. A. Stankovic, "Cityguard: A watchdog for safety-aware conflict detection in smart cities," in *Proc. 2nd Int. Conf. Internet-of-Things Design Implementation*, 2017, pp. 259–270.
5. M. Ma, J. A. Stankovic, and L. Feng, "CityResolver: A decision support system for conflict resolution in smart cities," in *Proc. 2018 ACM/IEEE 9th Int. Conf. Cyber-Phys. Syst. (ICCPS)*, pp. 55–64. doi: 10.1109/ICCPS.2018.00014.
6. E. Bartocci, J. Deshmukh, A. Donzé, G. Fainekos, O. Maler, D. Ničković, and S. Sankaranarayanan, "Specification-based monitoring of cyber-physical systems: A survey on theory, tools and applications," in *Lectures on Runtime Verification*, E. Bartocci and Y. Falcone, Eds. Cham: Springer-Verlag, 2018, pp. 135–175.
7. M. Meiyi, E. Bartocci, E. Lifland, J. Stankovic, and L. Feng, "A novel spatial-temporal specification-based monitoring system for smart cities," *IEEE Internet Things J.*, early access, 2021. doi: 10.1109/JIOT.2021.3069943.
8. M. Ma, J. Stankovic, E. Bartocci, and L. Feng, "Predictive monitoring with logic-calibrated uncertainty for cyber-physical systems," 2020, arXiv:2011.00384.
9. Y. Gal, "Uncertainty in deep learning," M.S. thesis, Dept. Eng., Univ. Cambridge, Cambridge, MA, 2016.
10. M. Ma, J. Gao, L. Feng, and J. Stankovic, "STLnet: Signal temporal logic enforced multivariate recurrent neural networks," in *Proc. Adv. Neural Inf. Process. Syst. 33 (NeurIPS)*, 2020.