International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

Integrated Multi-Sensor Fusion with Machine Learning for Real–Time Wastewater Pollutant Identification

***Shaik Ayisha, **D.Srinivasulu, ***Nyamathulla Shaik**
*PG Scholar, Kandula Lakshumma Memorial College of Engineering for Women, Kadapa, India, modem.priyankaa@gmail.com
**Asst.Prof, Kandula Lakshumma Memorial College of Engineering for Women, Kadapa, India, hodcse@klmcew.ac.in
**Asst.Prof, Kandula Lakshumma Memorial College of Engineering for Women, Kadapa, India, nyam.tisinfo@gmail.com

ABSTRACT

The reliable identification of pollutants in diverse environments (e.g., water, air, and sewers) is critical for the protection of human health and the anticipation of hazardous situations. Many of the existing studies analyzing data collected from sensors employ traditional machine-learning techniques. This article describes a new approach that includes two major contributions. The first is a deep-learning classification method that identifies pollutants in wastewater based on the conversion of raw data into meaningful natural-language metadata. The second is a low-cost data acquisition, pre-processing, and transmission hardware platform to support the deep- learning analysis. Overall, the proposed system delivered significantly better performance than leading edge state- of-the-art systems and will provide adequate computational efficiency. A major limitation of the proposed system is that it is reliant on the timing of the injection of the pollutant, which is defined as the exact time at which the pollutant enters the wastewater. To help overcome this limitation, an implementation of a finite- state-machine based timing mechanism will improve the reliability of the proposed wastewater analysis solution. There is an extensive presentation and discussion of the complete system. In addition, we provide many versions of the suggested processing method to evaluate how the system's promptness and computing load are affected by the amount of samples utilized. While the best baseline approach only managed an accuracy of 83.0%, our strategy achieved a minimum of 94.4%.

KEYWORDS: Multi-sensor Fusion, Machine Learning, Real–Time Monitoring, Low-cost sensors, water contamination and causal models.

I. INTRODUCTION

The need for precise environmental monitoring is a critical problem that will only grow in the coming years on a global scale. Numerous factors pertaining to the air,
soil, and water quality need regulation [1, 2]. Actually, by constantly keeping an eye on them, we can quickly and precisely restore ideal circumstances.
in response to potentially harmful occurrences, such the introduction of contaminants. Wastewater (WW) monitoring is crucial in this regard [3]. Wastewater (WW) is water that has been treated in order to make it safe to return to the natural cycle after being utilized for human or industrial purposes. The filtration systems need to know in advance what kinds of contaminants are in the water for them to work properly. Consequently, a water purification plant designed for domestic usage will vary from an industrial water purification system.
Therefore, processes to quickly identify incompatible compounds are very necessary to ensure the efficient and accurate functioning of purification facilities [4].The present method for resolving this issue involves the use of specialized laboratory equipment by the relevant regulatory agencies to conduct periodic monitoring operations at specific places along the water course. Despite its effectiveness, this approach lacks transparency on the water quality between successive tests and may not be conducted often enough to detect issues in a timely manner. The best course of action would be for the control

institutions to do occasional manual inspections in addition to the automatic distributed early warning monitoring. A distributed and continuous monitoring system can be installed with ease and at a low cost by using low-cost, Internet of Things (IOT)-ready systems [5]. These systems can gather environmental data and process it using centralized data collection and elaboration points.

Contamination detection and prediction from wastewater (WW): In literature, data analysis from instrumentation (sensor) systems for WW contamination detection is often done via some kind of traditional machine learning method [e.g., Decision Tree / Random Forest] (i.e. [6][7]); thus, we are proposing a new/different way (i.e. deeper learning) to accomplish this function. We are creating a unique detection system via the SENSIPLUS sensor network [Sensichips Srl, Pisa, Italy] [8] and using a causal generative ML architecture (originally developed for natural language processing) to create a more interpretable, organized structure of complex sensor data. For example, while commonly accepted methods (e.g.) for collecting data from one to another use data communication protocols (e.g. MQTT, message queuing, etc.) - the design of the actual transportation of said data is outside the bounds of this work. With the assistance of a partnership with Sensichips S.r.l., a set of data was created and eventually made available to others in the academic research community. Several cutting-edge models [9] were used as a baseline for performance comparisons of the new classification methods generated. The experimental data shows that the authors' approach consistently excels over the compared techniques. The experimental results further support the authors' methodology as practical and effective when applied in real (in situ) situations, both in terms of its robustness and effectiveness.

II. RELATED WORK

A lot of people in the scientific community talk about how important it is to monitor wastewater. To be more specific, sensors that can detect and categorize unwanted compounds are being developed using a variety of technologies in order to maintain an optimum water quality level. In order to keep tabs on air and water, a few writers used the SENSIPLUS platform to create monitoring systems [10, [11], [12], [13]. The results of the monitoring might range from a detailed categorization of the pollutants to a simple yes/no answer about the overall presence of contaminants. Instead of creating a general monitoring system that can function well in a broad variety of circumstances, it is frequently more preferable to design precise solutions for individual issues. Lim [14] outlines a pollution detection system in the WW framework, for instance, but it doesn't differentiate between compounds and the technology seems antiquated. Lepot et al. [15] takes a different tack, instead deploying an infrared camera to keep tabs on any illicit connections in the sewage system. Without worrying about the difference between chemicals, Ji et al. [16] provide an image processing approach that is meant to estimate the WW quantity. Even though the systems are durable against corrosion and provide an option for monitoring, cameras incur a significant energy cost to capture images, making both the system's cost and energy use limitations not considered in many studies. For example, Pisa et al. [17] developed a device that detects ammonium and total nitrogen, which extends a developed system that identifies & teaches nitrogen containing compounds. A new, portable device for monitoring wastewater pump station pump behavior and notifying of abnormal operating conditions has recently been developed by Drenoyanis et al. [18]. While the system is quite innovative in its design, as acknowledged in the original research, it unfortunately lacks a means of identifying or categorizing pollutants. Our present work is, to our knowledge, the first use of natural language processing techniques - more specifically, the use of causal models created for natural language generation - to detect contamination of wastewater from a data- processing standpoint. Many previous studies have established that language models can be successfully modified in order to solve problems outside of their conventional text-processing applications. For instance, in the medical field, many researchers have modified language models to accomplish two related goals: after reverse encoding (i.e., transforming coded data into descriptive text), to develop new rules for diagnosis [22] and to classify diagnostic tests [19-21]. Furthermore, false positive human mobility prediction methods have also used similar approaches [23][24].Transformers, which were first introduced as deep learning algorithms primarily for natural language processing (NLP) tasks [25], have achieved great

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

results in many different application domains outside of NLP. Some examples include: Image analysis [26] Video processing [29] Audio recognition [32] Conversational Agents [34] Recommender Systems / recommendation engines/ recommender algorithms [36] Reinforcement learning (RL) algorithms [38] Graph Based Modeling [40] Predicting protein structures [42] Autonomous driving systems [44] Anomaly Detection [46] etc. These successes show how well models created from the tools and techniques of NLP can be generalized to solve difficult problems found in other areas of everyday life or work environments [25].

III. SYSTEM OVERVIEW

The proposed solution is a complete end-to-end system comprising both hardware and software components. Each of these components and their respective roles are described in the following sections.

TABLE 1. Sensors used in the experiments.

IDE	Acquisition Frequency	Integer Value Proportional to
Platinum	78 kHz	Resistance
Gold	78 kHz	Resistance
Platinum	200 Hz	Resistance
Platinum	200 Hz	Capacitance
Gold	200 Hz	Resistance
Gold	200 Hz	Capacitance
Copper	200 Hz	Resistance
Copper	200 Hz	Capacitance
Silver	200 Hz	Resistance
Silver	200 Hz	Capacitance
Nickel	200 Hz	Resistance
Nickel	200 Hz	Capacitance

FIGURE 3. The software chain at a glance.

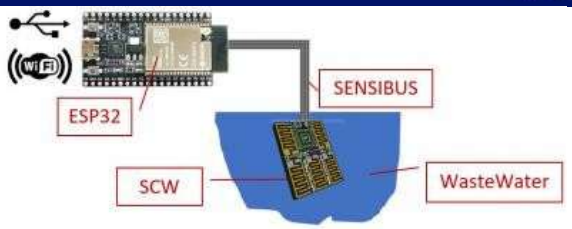


FIGURE 1. The hardware acquisition chain at a glance.

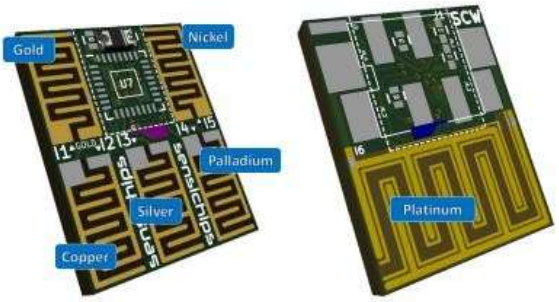


FIGURE 2. Smart Cable Water with its Inter Digitated sensors.

A. HARDWARE ARCHITECTURE

The developed setup comprises a fully integrated platform consisting of hardware and software modules to form complete systems as described in subsequent sections of this document. The data acquisition hardware includes the following components (Figure 1): a Sensor Module, a Smart Water (SCW), and a proprietary (single wire) sensor cable called SENSIBUS that enables the use of SCW for interaction and control. In addition to these components (sensor module and SCW), there is also a microcontroller unit (MCU) with firmware designed to monitor the SCW operational status and collect measurement data for transfer to a cloud platform.

The SCW allows the performance of both Electrochemical Impedance Spectroscopy (EIS) and voltammetric measurements. Sensichips s.r.l. holds patents and proprietary manufacturing techniques developed in conjunction with the University of Pisa to produce a low- cost, multi-sensor platform created using SCW technology. The SCW device effectively combines six IDEs, which are constructed from copper and coated with metals such as gold, copper oxide, platinum, silver, nickel, and palladium, and multiple sensing/chemical sensors for measuring SCW as shown in figure 2. The IDEs are coupled together to form the sensor array utilized for data acquisition and made from hydrocarbon materials that have been integrated into an embedded EIS circuit chip. To date, five of these IDE-based sensors have undergone experimental evaluation with respect to their performance. The IDE electrodes coated with platinum/gold were tested at both 200 Hz and 78 kHz. The IDE electrodes coated with silver/nickel were only tested at the lower frequency (200Hz).Since this parameter determines how the

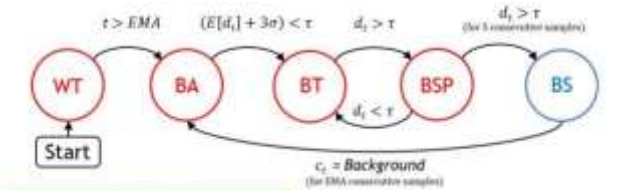
International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

metals on the sensors interact with the contaminants, changing the stimulus frequency permits the use of various frequency responses. Twelve different amounts DEs were obtained in a proportionate manner based on their capability and resistance. In Table 1, we can see how the frequencies and IDEs are related.

Section B: Software for Acquisition and Pre-Processing Figure 3 shows the software parts of the elaboration chain, which contain the following components: Classification system, C API firmware for the microcontroller unit (MCU), and Finite State Machine (FSM) for baseline acquisition and drug injection detection. In order to acquire raw data and transmit it to the computational module, the software of the MCU uses the SENSIBUS channel to control the SENSIPLUS chip. The FSM and classification system are controlled by the computational module, which may be a desktop computer linked by USB or a cloud-based system linked through TCP/IP and Wi-Fi.

Sections V–D detail the categorization scheme, whereas the following explains the FSM.

Figure 4 depicts the FSM, which operates in two stages: To normalize raw data, the FSM performs two steps: first, it extracts a baseline signal. Second, it selects whether to send each sample to the classifier and, if so, at what time.



Symbols	
t	time sequence
d_t	distance of s_t from b_t
EMA	Exponential Moving Average coefficient = 25
τ	the threshold (0.05)
c_t	classification of sample t
$E[d_t]$	the mean over last EMA
σ	the std dev over last EMA sample

States		Output
WT	Wait	BKG
BA	Baseline Acquisition	BKG
BT	Baseline Tracking	BKG
BSP	Baseline SusPended	BKG
BS	Baseline Stopped	The sample is forwarded to the classification module

FIGURE 4. Finite State Machine.

The FSM generates the baseline signal $\boldsymbol{b}_t$ by an Exponential Moving Average (EMA):

$$\boldsymbol{b}_t = \begin{cases} \boldsymbol{s}_t & t = 0 \\ \boldsymbol{b}_{t-1} & t > 0, S \in \{BS, BSP\} \\ \alpha \boldsymbol{s}_t + (1-\alpha) \cdot \boldsymbol{s}_{t-1}, & t > 0, S \in \{WT, BA, BT\}, \end{cases} \quad (1)$$

$$\boldsymbol{f}_t = \frac{\boldsymbol{s}_t}{\boldsymbol{b}_t}, \quad (2)$$

as shown in Equation 1, In this expression, sts_tst refers to the raw sensor data, and btb_tbt refers to the baseline signal. The vectors ft\mathbf{f}_tft, st\mathbf{s}_tst, and bt\mathbf{b}_tbt are all nnn-dimensional vectors that are combined element-wise in the expression, where nnn refers to the total number of sensors (see Table 1).

With the inclusion of the baseline signal, the

$$\boldsymbol{d}_t = \|\boldsymbol{f}_t - \boldsymbol{u}\|. \quad (3)$$

system is able to counteract external disturbances, transient signal spikes, sensor drift, and inter-sensor differences. As shown in Figure 4, the finite state machine (FSM) runs with ttt referring to the current time index, τ\tauτ referring to a threshold value set to 0.05 in the experiment, and dtd_tdt referring to the Euclidean distance between the feature vector ft\mathbf{f}_tft and a reference vector u\mathbf{u}u, where u\mathbf{u}u is an all-ones vector.From Equation (2), the feature vector ft\mathbf{f}_tft is a unit vector when the baseline signal btb_tbt is equal to the sensor signal sts_tst. In this scenario, the unit vector is used to calculate the Euclidean distance, and a distance value of dt=0d_t = 0dt=0 means that the baseline signal btb_tbt is tracking the sensor outputs sts_tst exactly.

The WT state serves to initialize and fill the exponential moving average (EMA), after which the system automatically switches to the BA state after acquiring EMAc\text{EMA}_cEMAc samples. In the BA state, the finite state machine (FSM) starts tracking the incoming signal and waits for a stable alignment with the baseline to be reached. Reliable tracking is determined by examining the distance between the signal and the baseline.

When the variability of this distance, computed as the mean plus three standard deviations, goes below a certain threshold τ\tauτ, experimentally set to 0.05, the system switches to the BT state. If the instantaneous distance exceeds τ\tauτ, the system checks whether an external substance has been introduced into the monitored environment. Upon observing a deviation of the signal from the baseline, the FSM temporarily switches to the BSP state.

To avoid material release events from being mistakenly identified as temporary noise or isolated measurement spikes, the FSM switches back to the BT state if the distance fails to persistently exceed the threshold for five consecutive samples after entering the BSP state. After the FSM enters the BS state, the current normalized feature vector ft\mathbf{f}_tft is submitted to the classification module for further processing.

Part C: The Module for Classification

A designed classification component is based on deep learning approaches to natural language processing, making use of a model within the Transformer framework [48]. The model choice for this work was T5, a very large text-to-text architecture that has been trained to support the text-to-text workflow through a combination of supervised and unsupervised training tasks. The supervised component consists of tasks aimed at converting structured text data into other types of structured text data, while the unsupervised components included denoising text or learning representations for the text.

The T5 architecture is built following the typical architecture of the Transformer framework. As a result, the architecture contains sequentially stacked encoder blocks for mapping text input to latent features, as well as decoder blocks for mapping latent features back to output text. The encoder and decoder blocks use self-attention, encoder- decoder attention, and feed-forward mechanisms in their respective layers. As a generative language model, T5 generates text by taking in text input and producing text output.

For the purposes of this study, the T5 model will be trained to generate labels to identify the specific pollutants present in the waste water samples tested. Therefore, prior to fine- tuning the T5 model to perform that function, the sensor signal will first be processed and formatted into an input appropriate for the T5 model to use during classification; then the pre-trained knowledge of the T5 model will be fine-tuned for the purpose of identifying those substances. For further details about the classification framework, please see Section V of this report.

IV. DATASET

This section describes the procedure used to collect the datasets.

Section A: Chemicals

The Pollutants in WasteWater (WW) are the concern of this research study's dataset, which aims to identify the chemical compound spillage that may pose a threat to the public or impair the functionality of purification systems. To achieve efficiency, the acquisitions were conducted in a controlled lab setting. The biological danger posed by the presence of unknown microorganisms or contaminants and the requirement for a controlled measurement environment resulted in the elimination of measurements conducted at experimental sites. In reality, meteorological phenomena such as rain alter the WW composition over time. The samples were obtained from two different laboratories situated in Italy and Poland from 2019 until 2021, and a summary of the chemical utilized in the experiments can be found in table two. The following sections will outline how the data were created (using 10 measurements of each chemical). In addition to the 10 measurements of each chemical, there were also measurements taken of the background and wastewater samples.

Part B: The Dataset

A dedicated measurement configuration was used to obtain the experimental data for this study. The experimental measurement configuration is comprised of a personal computer (which serves as the primary control unit), a microcontroller (which serves as the communication link between the computer and multi-sensor platform), and a Solid State Cable Water (SCW) reader (which is responsible for recording and storing the actual measurements from the sensors). A beaker containing 300 ml of WW has been loaded with the different chemicals, and the SCW is placed inside it. The same conditions were maintained for each measurement run by simulating the motion of the WW using a magnetic stirrer. In an effort to avoid the possibility of air bubbles, which could cause noisy data, the 25 mm long anchor was designed to rotate at 50 rpm (turbulent flow). The samples of the dataset were measured in two steps, each using a different measurement procedure:

The measurement process was broken down into two components. The first component was a warm up phase, where 600 wastewater samples were collected. The second component was to collect 1,000 additional samples after the addition of a target drug. Each of these two component's were applied to a total of 10 trials for each material. Each sample acquisition had a typical interval of approximately 1.6 seconds giving a total of 1,600 samples per chemical for each measurement cycle. This resulted in approximately 40 minutes of total acquisition time for each experiment. An example of this can be seen in (Figure 5).

V. EXPERIMENTAL ANALYSIS

A. Problem Formulation

Measurements collected from ten different

chemicals included with the controls (i.e., wastewater). K-fold cross- validation contributed to reliable and unbiased evaluation allowing for stable results, less variability due to random chance, and less biases. In particular, we use a 10-fold validation procedure, in which we use the remaining experiments for training and rotate the one utilized for testing from 1 to 10. Using this method, we can be confident that each experiment's samples will only be tested once, and that we can average the performance metrics over all 10 test sets. Because of this, we can get a solid assessment of how well our models work. In the "warm up" phase, the monitoring sensors received only wastewater as a test, and throughout that test they collected 600 samples for each drug. After injection, 1,000 more samples were collected while the sensors stayed in constant contact with both wastewater and the administered drug (if there was one).In accordance with the k-fold method described earlier, a training set of nine acquisitions was used for each chemical (including WW), whereas a test set of one acquisition was utilized. By averaging the test acquisitions, we may assess the models' efficacy, and improved using the Reduced Error Pruning Tree algorithm as compared to that of using C4.5 as an example [54].Random Forest [52] is the algorithm we've used for decision trees. The instance-based algorithms have chosen the traditional k-nearest neighbours technique (KNN) [55], the kernel-based algorithms have chosen Support Vector Machines (SVM) [56], and the category algorithms of Artificial Neural Networks have chosen a Multi Layer Perceptron (MLP) [57]. In the end, ensemble-based algorithms such as MLP, KNN, SVM, and Random Forest were chosen by a majority vote

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

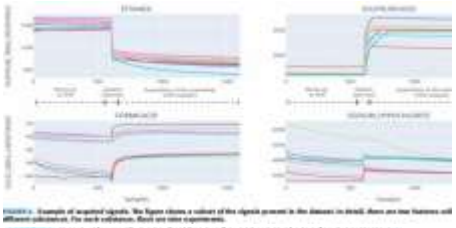


TABLE 3. Number of samples by substance. For each acquisition, 600 samples are collected in Wastewater and 1000 samples after substance injection. The training set of each fold is composed of 9 acquisitions for each substance, while the test set comprises of 1 acquisition.

Substance	Training set	Test set
Wastewater	(600·10·9 + 1000·9)	(600·10·1 + 1000·1)
Acetic acid	(1000·9)	(1000·1)
Acetone	(1000·9)	(1000·1)
Ammonia	(1000·9)	(1000·1)
Ethanol	(1000·9)	(1000·1)
Formic acid	(1000·9)	(1000·1)
Hydrochloric acid	(1000·9)	(1000·1)
Hydrogen peroxide	(1000·9)	(1000·1)
Phosphoric acid	(1000·9)	(1000·1)
Sodium hypochlorite	(1000·9)	(1000·1)
Sulphuric acid	(1000·9)	(1000·1)
Total samples	158,400	17,600

Only on the one thousand samples that were gathered after the warm-up period. You may get more information on the dataset folds in Table 3.

Baseline Methods to evaluate how effective the suggested method is, the dataset was assessed towards using a predetermined number of baseline learning algorithms that are commonly used in machine learning. The intent of choosing these methods was to allow for wide-ranging comparisons amongst systems with different complexities. Therefore, we used methods from a variety of categories: boosting, bagging, tree-based, instance-based, kernel- based operating on artificial neural networks, and ensemble classifiers. The method we used was to implement some variants of the BOOSTING method AdaBoost [50] with near a dozen different types of weak learners, including (1) decision stumps; (2) J48 decision trees [51]; and (3) The Random Forest Classifier [52] (which is one of the more advanced types of classifiers available). In the case of bagging, Reduced Error Pruning Tree was also used and derived using a light-weight decision tree algorithm that creates splitting attributes based on information gain (in two binary over numeric features). What makes Reduced Error Pruning Tree a viable candidate for making use of the advantages of the bagging method is because the computational complexity of using and training a model to evaluate accuracy is much To get optimal models through a methodical search and trials of hyper parameters, we completed a systematic grid search on several selected machine-learning algorithms. A summarized presentation of all the tuned hyper parameter settings for each of the algorithms is shown in Table 4. All experiments in this study used [58] WEKA (Waikato Environment for Knowledge Analysis, version 3.8.6) for implementing the algorithms.

A feature vector is used as the input data to the learning algorithms, and it is calculated as shown in Equation 2. The feature vector represents the outputs from all of the sensors listed in Table 1 (converted to real-time measurements that were normalised against the baseline measurements from each sensor). As the models used in this study (both benchmark and the method proposed herein) were trained on the same dataset using the same feature representation, this guarantees that all variation in the results produced from the models is caused solely by the modelling methodology (model type) rather than inherent discrepancies in either any of the datasets used or in the input variables being inputted to either of the models.

VI. RESULTS AND ANALYSIS

A. Performance

The performance metrics for both the baseline approaches and our new model are summarized in Table 5, calculated as averages across all test sets. Because the new method can process multi-terabyte timestamp data (see Section V- D), performance values obtained by sampling one, two, five, ten and one-hundred terabyte timestamps to aggregate model outputs are reported below the table. Even with the most restrictive condition on using only one terabyte timestamp (T5, one sample), all models tested perform better than current methods on each evaluated variable. This result is illustrated in Figure 6, which shows weighted F1 (F1W) values for all models tested across all test datasets. The x-axis shows model names, the y-axis shows their corresponding F1W values, and the strongest current model's performance is shown

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

as a horizontal dashed line. For each model, there are three bars which give F1W values from the three folds used in each cross-validation. All other evaluation metrics, not shown here, exhibit the same trend.

TABLE 5. Average effectiveness metrics computed over the test sets. $metric_M$ denotes the micro-averaged metric, $metric_m$ the and $metric_W$ the weighted metric.

Model	Acc	MCC	$Prec_M$	$Prec_m$	$Prec_W$	Rec_M	Rec_m	Rec_W	$F1_M$
AdaBoostM1_DecisionStump	0.418	0.208	0.418	0.004	0.192	0.418	0.111	0.418	0.418
AdaBoostM1_J48	0.802	0.767	0.802	0.720	0.744	0.802	0.732	0.802	0.802
AdaBoostM1_RandomForest	0.767	0.722	0.767	0.672	0.705	0.767	0.673	0.767	0.767
Bagging	0.764	0.718	0.764	0.687	0.720	0.764	0.672	0.764	0.764
knn	0.803	0.769	0.803	0.732	0.746	0.803	0.730	0.803	0.803
MLP	0.794	0.758	0.794	0.724	0.745	0.794	0.715	0.794	0.794
RandomForest	0.779	0.737	0.779	0.682	0.711	0.779	0.693	0.779	0.779
SVM	0.802	0.760	0.802	0.749	0.754	0.802	0.726	0.802	0.802
Vote	0.810	0.778	0.810	0.747	0.756	0.810	0.739	0.810	0.810
T5 (1-sample)	0.915	0.865	0.915	0.950	0.950	0.915	0.950	0.950	0.915
T5 (2-samples)	0.914	0.864	0.914	0.936	0.936	0.914	0.936	0.936	0.914
T5 (5-samples)	0.927	0.877	0.927	0.958	0.958	0.927	0.958	0.958	0.927
T5 (10-samples)	0.958	0.908	0.958	0.976	0.976	0.958	0.976	0.976	0.958
T5 (50-samples)	0.944	0.894	0.944	0.958	0.958	0.944	0.958	0.958	0.944
T5 (100-samples)	0.975	0.925	0.975	0.988	0.988	0.975	0.988	0.988	0.975

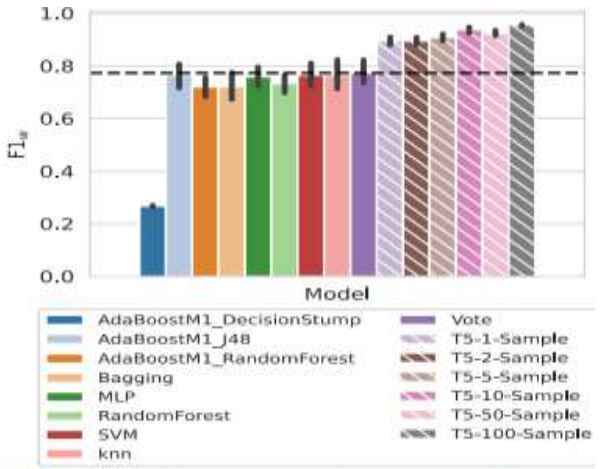


FIGURE 6. Average effectiveness metrics computed on the test sets for $F1_W$. The dotted line represents the best baseline method.

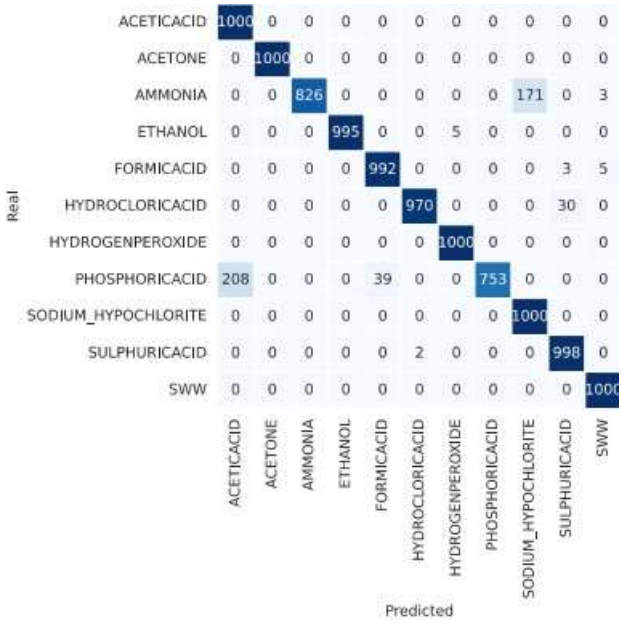


FIGURE 7. Distribution of the real and predicted values for the T5 (100-samples) model.

In the last row of Table 5 there is a confusion matrix for the best performer which is the T5 model trained on 100 data points and the corresponding result is shown in Figure 7. The confusion matrix details how the model predicted classes on the test dataset that had been classified into distinct chemical classes (the chemical classes are delineated in the matrix row) and the model's prediction for each class (the chemical classes the model predicted are delineated in the matrix column). There appears to be an extremely high degree of accuracy for the prediction of each substance from the model; the only exception being ammonia (which was predicted as sodium hypochlorite 171 times of 1000) and phosphoric acid (which was predicted as acetic acid 208 times out of 1000). The high level of accuracy for each of these chemicals is likely due to the fact that they have very similar response patterns from the sensors including how they respond during the time period prior to chemical injection.

The lower half of Table 5 illustrates the apparent positive correlation between the model's performance and quantity of pre-injection timestamps (tb) used in an input. In addition, performance metrics become better as the quantity of baseline timestamps increases. Figure 8 reveals this correlation visually; the x-axis contains the number of timestamps (tb) provided as inputs and the y-axis contains the associated model performance scores. Based on these observations, providing later model training with more data during a warm-up phase allows models to have greater predictive reliability because they have more exposure to different types of early stage signal variation (or "early" chemical signatures) used to differentiate between chemical signatures prior to injecting.

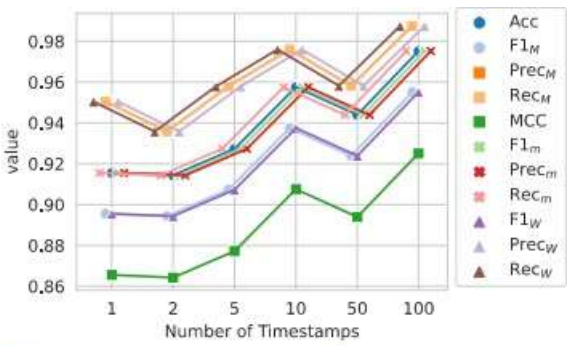


FIGURE 8. Average effectiveness metrics obtained when varying the number of timestamps t_b fed to the T5 model.

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

B. EFFICIENCY

The NVIDIA GeForce RTX 3090 was used for evaluating inference performance. Without batching, the system's average time to make one prediction for one timestamp (the time it takes to get through a complete generation of the full text using beam search with an early stop) is approximately 76 ms (see Section V for full details of the inference process). This calculates out to 76 seconds to produce an example for 1,000 timestamps giving the system an output of approximately 13 predictions per second. Utilizing batched inference will provide additional benefits by allowing the processing of larger batches (greater than 512 samples) and, therefore, providing a significant reduction in the time to make predictions. In addition, the CPU inference performance was assessed on 16 Intel® Core™ i7-10700 processors with a clock speed of 2.90 GHz; the average time to process one prediction is approximately 450 ms. The results show that the model developed for this study is able to perform real-time predictions in both CPU-only and GPU-accelerated environments. When validating the proposed method, it is also necessary to consider the computational requirements associated with training as well as inference efficiency as both affect the overall feasibility of the deployed model. Causal Language Models such as T5 are typically made up of millions to billions of parameters; therefore, they require an immense amount of computational and data resources to optimize. Training of these models is performed via the standard transformer architecture with computational complexity $O(N^2d)$, where N is the input sequence length and d is the hidden dimension of the model[59]. These results in linear increase in the cost of training based on increasing size of the models hidden size and quadratic increase in cost based on the increase in length of the input sequences. Due to this high computational cost of training, fine-tuning pre-trained models allows for their adaptation for certain tasks with the use of much smaller task-related datasets. Therefore, it is possible to effectively re-use pre- trained models for various applications. Furthermore, since the vast majority of pre-trained models have already been optimized on very large data sets for general language modeling, fine-tuning pre-trained models can be accomplished much more efficiently and cost effectively than training a model from scratch. In general, when fine-tuning a pre-trained model, the number of epochs will be limited to three or fewer, although this may vary depending on the size of your task-specific data set. How long it takes to fine-tune a pre-trained model will vary based on both the size of your data set as well as the amount of processing power at your disposal; fine-tuning may take anywhere from just a few minutes to many hours depending on these factors. We used the Hugging Face package to get the model's weights for this work and spent around an hour on the GPU architecture mentioned before fine-tuning the recommended strategy. Once adjusted, the model has endless potential for the job at hand, as detailed in this article.

VII. DISCUSSION AND CONCLUSIONS

A study was conducted evaluating the ability of various NLP models to detect pollution in wastewater. One focus of our research was on developing generative causal models, particularly T5, as a generative machine learning approach. We defined a new method of generating input features called Natural language representation, where characteristics of input features would be transformed into textual format prior to application to a generative language model. This allows the model to classify and analyze input samples based on whether they contain pollutants or not. The suggested technique was subjected to an experimental evaluation in which its efficacy was quantified and compared to a collection of state-of-the-art baselines. The suggested approach is more efficient and effective than the baseline methods, and the experimental results demonstrate this. This makes it suitable for deployment and real-world use.

We will explain why this unconventional strategy makes sense and really works in reality, despite the fact that it may seem unusual or counter-intuitive at first. Recent studies have shown that attention-based models and transformers can generalize to a wide range of tasks, even ones that the model has never been trained on. This

includes tasks that are unrelated to or not expressed naturally using NLP, such as images, videos, graphs, and reinforcement learning. Because of its attention mechanism and almost task-agnostic training approach, transformer-based models are able to generalize. Essentially, this method is about reconstructing parts of a given input element that is either masked or corrupted; the same reconstruction also applies to when an element is missing (for example, at the end of a series of input tokens) and needs to be predicted using the general/undefined data provided by the other parts of the input. By combining these two tasks, the model learns the meaningful, and more importantly general, latent relationships within an input dataset, which in turn provide information that will help determine what, will be produced by the output of the given input dataset. An example of this can be found in how the model generates new text from a set of text inputs; or how it completes or reconstructs a damaged or incomplete image or frame of an image from multiple frames; or how it learns complex structures within the data used in a graph, including the relationships between the nodes of the graph and the edges of the graph. Additionally, transformer-based models that were trained to complete or generate new output based on input using a masked/casual form of the input have been shown to be highly applicable to tasks and domains beyond the examples. That is why we think our neural network can make reliable predictions about the potential pollutants in wastewater based on the textual descriptions collected from the sensors.

There are a few caveats that must be mentioned, even if our method did provide some encouraging findings. The suggested method depends on knowing when to inject the contaminating compounds, which is one of its major drawbacks. This implies the algorithm could struggle to

correctly categorize wastewater pollutants in the absence of knowledge about the injection timing. Using a finite state machine which can precisely determine injection time we were able to resolve this problem in our work. However, we want to address this constraint and implement a unified system in our future studies by focusing on integrated method development.

A key constraint of this proposed technique is its dependence on a sufficient amount of labeled training data (for Construction of a DL training set). It is often difficult to ensure enough labeled training data is available due to the need for access to pollutant/contaminated wastewater samples in practice.The data scarcity problem will be addressed in future research by exploring other methods of synthetic data generation or by looking at transfer learning approaches.

This study paves the way for future research into using language models for environmental tasks like pollutant analysis and detection. The explicit goal of our future efforts is to promote the widespread use of models based on natural language across several areas and activities pertaining to drug detection. Additionally, it will use zero and few-shot learning procedures and interpretability frameworks to explore the model's generalization and explanation skills.

REFERENCES

[1] L. T. Lee and E. R. Blatchley, ‘‘Long-term monitoring of water and air quality at an indoor pool facility during modifications of water treatment,’’ Water, vol. 14, no. 3, p. 335, Jan. 2022. [Online]. Available: https://www.mdpi.com/2073-4441/14/3/335

[2] H. Chojer, P. T. B. S. Branco, F. G. Martins, M. C. M. Alvim-Ferraz, and S. I. V. Sousa, ‘‘Development of low- cost indoor air quality monitoring devices: Recent advancements,’’ Sci. Total Environ., vol. 727, Jul. 2020,Art. no. 138385. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0048969720318982

[3] K. Farkas, L. S. Hillary, S. K. Malham, J. E. McDonald, and D. L. Jones, ‘‘Wastewater and public health: The potential of wastewater surveillance for monitoring COVID-19,’’ Current Opinion Environ. Sci. Health, vol. 17, pp. 14–20, Oct. 2020.

[4] A. Trubetskaya, W. Horan, P. Conheady, K. Stockil, S. Merritt, and S. Moore, ‘‘A methodology for assessing and monitoring risk in the industrial wastewater sector,’’ Water Resour. Ind., vol. 25, Jun. 2021, Art. no. 100146.

[5] E. Syrmos, V. Sidiropoulos, D. Bechtsis, F. Stergiopoulos, E. Aivazidou, D. Vrakas, P.

Vezinias, and I. Vlahavas, ‘‘An intelligent modular water monitoring IoT system for real-time quantitative and qualitative measurements,’’ Sustainability, vol. 15, no. 3, p. 2127, Jan. 2023.

[6] D. Jalal and T. Ezzedine, ‘‘Decision tree and support vector machine for anomaly detection in water distribution networks,’’ in Proc. Int. Wireless Commun. Mobile Comput. (IWCMC), 2020, pp. 1320–1323.

[7] D. G. Eliades and M. M. Polycarpou, ‘‘Water contamination impact evaluation and source-area isolation using decision trees,’’ J. Water Resour. Planning Manage., vol. 138, no. 5, pp. 562–570, Sep. 2012.

[8] A. Ria, M. Cicalini, G. Manfredini, A. Catania, M. Piotto, and P. Bruschi, ‘‘The SENSIPLUS: A single-chip fully programmable sensor interface,’’ in Applications in Electronics Pervading Industry, Environment and Society, S. Saponara and A. De Gloria, Eds. Cham, Switzerland: Springer, 2022, pp. 256–261.

[9] M. Molinara, C. Bourelly, L. Ferrigno, L. Gerevini, M. Vitelli, A. Ria, F. Magliocca, L. Ruscitti, R. Simmarano, A. Trynda, and P. Olejnik, ‘‘A new dataset for detection of illegal or suspicious spilling in wastewater through low- cost real-time sensors,’’ in Proc. IEEE Int. Conf. Smart Comput. (SMARTCOMP), Jun. 2022, pp. 293–298.

[10] M. Ferdinandi, M. Molinara, G. Cerro, L. Ferrigno, C. Marroco, A. Bria, P. Di Meo, C. Bourelly, and R. Simmarano, ‘‘A novel smart system for contaminants detection and recognition in water,’’ in Proc. IEEE Int. Conf. Smart Comput. (SMARTCOMP), Jun. 2019, pp. 186–191.