IoT-Enabled Predictive Analytics and Real-Time Monitoring: Applications in Disaster Management

S Soundarya

***Abstract*:** The integration of IoT devices in real-time data collection and predictive analytics provides transformative potential toward disaster forecasting and responses. However, existing research faces quite significant challenges including denoising effects, hotspot issues, inefficient resource allocation, overfitting of data, and complexity in decision-making. The proposed solution is a predictive analytic framework for IoT-enabled disaster management, aimed at overcoming these limitations. The methodology is initiated through the construction of a disaster management network by using IoT sensors to generate synthetic data. The collected dataset is denoised using the Denoising Autoencoder (DAE) to get rid of noise and improve data quality. In tuning clustering and resource allocation, we apply the Tuna-Swarm Algorithm (TSA), which aided in efficient management of IoT resources. Switchable Normalization in a custom-designed SN-Convolutional Neutral Network architecture is leveraged for the resources suffering from overfitting. Disaster prediction is based on the Mobile Net-Transformer Generative Model (MN-TGM), an advanced system designed for proper and timely forecasting. The decision-making and resource allocation tasks are subsequently simplified through the use of Fault Tree Analysis as well as Decision-Making Trial and Evaluation Laboratory methods (FTA-DEMATEL). The various performance metrics include latency (ms), overfitting rate (%), prediction accuracy (%), model accuracy (%), and energy efficiency (%) against cluster count. After simulating the system in Ns3.30.1 Ubuntu with Python, its robustness and effectiveness were confirmed. This framework presents a holistic solution to mitigate the existing challenges presented by IoT-enabled disaster management, ensuring timely interventions, exclusive allocation, and improved decision-making, thus averting disaster risks and their impacts.

Index terms:* *Disaster Management, IoT, prediction, resource allocation, overfitting and decision making.

I. Introduction

The Internet of Things (IoT) is a revolutionary and rapidly advancing technology which, perhaps, will change the manner in which we interact with our surroundings [1]. As technologies in communications, computing, and embedded systems have matured rapidly, IoT systems have been deployed increasingly in a wide range of application contexts [2]. In recent years we have facilitated the emergence of the Internet of Things (IoT) as one of the fastest-growing sources of data [3]. Disaster management is one of the many application areas where recent developments in IoT technology are helpful. Because it is so common and has the potential to save lives, the Internet of Things is essential to disaster management. Even if disasters cannot be predicted with precision, damage can be minimized with prompt relief efforts. Planning for disaster management is mostly dependent on the region's topography, climate, habitat, etc., as well as the resources that are accessible. Emergent IoT networks, in which objects (sensor nodes) are linked to one a different one, have recently been used to construct real-time monitoring and catastrophe warning systems. One notable example of this being a significant component of IoT is wireless sensor networks (WSN), which are frequently used to monitor natural catastrophes in distant and inaccessible regions [4, 5, 6]. In the recent times, the wireless sensor network has become a popular option because of its simplicity and easy maintenance, price performance of sensors, and high data accuracy, since the wireless sensors do not normally need much power, leading them to exist longer [7]. Smart sensors and actuators lack the capacity for large storage, thus allowing a small number of battery cells to be used for a smart connectivity in the IoT context. So, the data transmission process is crucial in the transition of data flow from the IoT layer to cloud-edge layer for processing and storage. Based on multiple research works, IoT is structured for treating safety-critical case studies like wildfires, earthquakes, flooding, blizzards, hurricanes, seasonal tornadoes, and landslides for the management of disasters [8]. Disasters are aberrant events that occur naturally or otherwise and lead to massive damages, destruction, and human suffering. Disasters causing these enormous losses around the globe, both economic and human, and the disaster response systems have grown [9]. These are just a few of the reasons why a Disaster Management System with IoT and AI capabilities remains one of the most important systems in the world [10]. Disaster management is the implemented holistic strategy that involves five elements prevention, protection, preparedness, emergency actions, and recovery [11]. DM includes the management of disaster risks and impacts. It comprises good riskiness, good resuscitation, good preparedness, and good recovery. It includes organization, command, and use of administrative resources against disasters. The practitioners of this domain are trying to lessen or avert the devastation from natural disasters and provide immediate help to the victims of the disaster and rapid recovery. Various other tasks provisioned the operationalization of this domain-rendered, including risk assessment, preparedness, mobilization, logistics, and housing or servicing [12]. Disasters such as these usually develop so fast that provides certain characteristics; these may include complexity, multiplicity, and extension of impact intensity, great destructiveness, and propagation of determinants of disaster in variable combinations [13]. There are various disaster events caused by an earthquake or an environmental hazard. Most disasters cause fatalities and tackle property damage [14]. Because of the high level of population density, lack of resettlement facilities and low capacities for hazard reduction from the adverse fates of nature on human lives, systems of buildings, roads, bridges, etc., developing countries are more prone to very high risk and have very little capacity to cope with them [15]. Reducing loss of human life in any kind of disaster is one of the five major objectives of any disaster management system. Mainly, lives are to be saved-the task remains the same regardless of whether a disaster is through the form of a natural disaster or through traffic accidents. It becomes highly significant to develop an efficient disaster management system (DMS) aimed at protecting lives during such times. High-quality images must be taken and supplied to the rescue operation to assist the DMS in its rescue work. Perhaps to obtain such data one needs the use of technological applications and devices [16]. Implied accordingly are two different concepts, disasters and threatening events to nowadays described as threats, and it is in this sense that disasters could be said to develop when humans are exposed, becoming defenceless against said threats or vulnerabilities to disasters [17]. However, the existing approaches with several node deployment processes assume uniformity, which is impractical. For certain practical applications, the nodes are placed in the network in almost no uniform pattern. Hence, clustering and routing would appear to be the ideal solutions to disaster management in IoT networks. The clustering process-Separates the nodes from the entire body of nodes within the capture area and groups them together in a generalized manner-creates a few clusters, every one of which is led by a leader referred to as the CH. All other nodes within a cluster are referred to as CMs. The CH is mostly accountable for collecting sensed data from its CMs and forwarding it to either the BS or multiple relay nodes. Additionally, disaster management, in fact, requires efficient routing protocols with random node deployment. Routing protocols provide the mean of delivering data. They determine what specific addressing information will be forwarded by the packet based on the characteristics of the internetwork. The routing procedures find best paths to destinations depending on some criteria like energy, distinctiveness, link quality, and so on. Since the choice of CHs was tedious and the route to be taken must be optimal, it can be categorized as an NP problem, which is solvable using meta-heuristic optimization techniques [18, 19]. However in the research aim is to utilize IoT-enabled predictive analytics and real-time monitoring for disaster management.

A. *Motivation & Objectives*

Disaster management has encountered several problems, such as uncertainty in data, misallocation of resources, and the necessity of efficient decision-making processes. An Innovative technology offering promising solutions but with serious limitations to be overcome for enhanced disaster response. The following are the main issues raised by existing works:

- **Impacts in Denoising:** The major issue of disaster management is noise-induced uncertainties due to noisy input data and irrelevant factors that significantly impact model accuracy. In the absence of proper pre-processing for removing noise and overfitting, disaster prediction will not be reliable and, thus, prevent effective decision-making.

- **Limitations in Hotspot Issues Resolution in IoT Networks:** The proposed method has limitations in resolving the hotspot issues within IoT networks. Although an unequal clustering mechanism is introduced, challenges persist in the effective management of IoT network performance in the context of disaster scenarios.

- **Inefficient Resource Allocation and Deployment:** Allocating and deploying resources based on disaster categories remains a limitation, as the current mechanisms do not always adapt efficiently to the rapidly changing demands of an emergency. This results in delays and inefficiencies in emergency management.

- **Data Overfitting Issues:** In existing methods faces challenges with data overfitting and the limited resolution of scalograms, which can degrade model performance and accuracy. While dropout layers help mitigate overfitting, the model's generalization still faces limitations that hinder optimal disaster prediction.

- **Need for More Advanced Machine Learning Algorithms:** Although the results from CNN and LSTM models are effective, there is a need for further exploration of more advanced machine learning algorithms, such as Transformer models, to enhance the accuracy and prediction capabilities for disaster management.

- **Challenges in Community Involvement and Decision-Making:** A significant limitation of IoES implementation is the challenge of addressing the diverse needs and preferences of various communities. Additionally, ensuring effective involvement of these stakeholders in decision-making processes remains complex and resource-intensive, impeding the potential for full IoES integration in disaster management.

The research goal is to utilize IoT-enabled predictive analytics and real-time monitoring for disaster management. The primary objectives are

- Develop a robust framework for data collection and pre-processing to address noise and ensure accurate predictions.
- Implement efficient clustering techniques to optimize resource allocation and resolve hotspot issues in disaster-prone regions.
- Enhance model generalization and prediction accuracy through advanced training techniques.

- Design hybrid models for precise forecasting of disaster events based on spatial and temporal data.
- Incorporate decision-making methods for a complete disaster risk analysis and effective real-time response.

B. Research Contribution

This research explores innovative data-driven solutions for improving disaster management by utilizing IoT-enabled predictive systems and real-time monitoring. It seeks to enhance disaster preparedness, minimize response times, and optimize resource allocation to mitigate the impact of natural and man-made crises effectively. The research highlights were given as follows:

- To Demonstrated the applicability of the Denoising Autoencoder (DAE) for pre-processing disaster data to eliminate noise and irrelevant factors, improving model reliability.

- A Tuna-Swarm Algorithm (TSA) is to optimize clustering with resource allocation in IoT-enabled disaster management networks.

- Then the Applied Switchable Normalization within (SN-CNN) framework to counter overfitting and enhancing generalization capabilities of the model for disaster scenarios.

- The Proposed MobileNet-Transformer Generative Model (MN-TGM) for advanced disaster prediction with lightweight architectures and attention mechanisms, which gives better feature extraction along with superior temporal insights.

- Finally Integrated fault tree analysis and DEMATEL (FTA-DEMATEL) methodologies for the fine-tuning of disaster risk evaluation and the decision-making process.

C. Paper Organization

The remaining portions of this work are structured as follows. The presented methodologies and literature evaluation are provided in Section II. In Section III, the issue with the description is supplied. The recommended research technique, including the protocol and algorithm, is described in Section IV. In Section V, simulation results are obtainable along with a comparison between the recommended strategy and existing approaches. The proposed method is explained in Section VI.

II. Literature Survey

This section summarizes and examines the major research gaps addressed in the previous works. The author of [20] describes an "ensemble learning approach based on Bayesian Model Combination (BMC-EL)" that incorporates IoT technology in predicting the flood depth in coastal cities. It introduces flood intensity classification and K-fold cross-validation in training subsets with the help of base models like BPNN and RF. Real flood data from Macao validates the reliability and accuracy of the BMC-EL approach in flood prediction through experiments. However their future research will apply more comprehensive datasets to the design and refinement of street flooding prediction models with higher precision. Information technology will advance so that richer and more diverse data concerning flooding and its causes can be collected. The author in [21] presents an IoT-based prototype to gather hydrological and meteorological data, such as water flow, level, temperature, discharge, humidity, direction and wind speed. Using the LSTM model, the collected data is analyzed to categorize flood actions into dissimilar alert levels. Additionally, a new method for defining water discharge is proposed, considering river flow, average depth and sectional area width. To further improve the accurateness of flood forecasting, the scheme could be combined with future remote sensing technologies and geographic information systems.

The author in [22] looks into the crucial role AI plays in the four phases of the disaster management process, tracking and mapping, recovery operation, and others. Here, it brings out a good integration of AI with GIS and RS for efficient planning, analysis, and situational awareness. Thereby, these technologies, being machine learning and geospatial analysis, help have quicker and more efficient hazard and disaster responses and enhance decision-making in real time. Furthermore Combining AI technology, large complex multispectral datasets, and Geographic Information Systems possess vast opportunities in preventing or reducing damages brought upon by natural disasters as well as man-made tragedies. This is because team success depends on more than merely managing information and analytical powers. Other integral technological components come in the picture that improves upon disaster team effectiveness. The authors of [23], presents two blocks: Block-I employs a CNN to sense and identify natural disasters and Block-II applies a distinct CNN for classifying disaster intensity levels using different filters and parameters. This approach enhances the accuracy of detection as well as disaster categorization. The blocks work in tandem to provide a full solution for disaster monitoring. However, the real-time data handling would face problems with the complexity of the model and high computationally demanding. This one depends upon the quality and kind of training data and has its performance limitations in scenarios varying. The classification accuracy is bound to reduce with very rare or unprecedented events like disaster.

The author of [24] presents the integration of autoencoder and convolutional neural network, 3S-AE-CNN, can estimate earthquake magnitude and location very fast within 3 seconds from the onset of the P-wave. The model, trained on seismic data from Japan's Hi-net network, accurately predicts earthquake parameters and transmits them via an IoT system for rapid disaster response. Compared to traditional methods, the 3S-AE-CNN model demonstrates improved accuracy in magnitude and location estimation, proving its effectiveness for Earthquake Early-Warning Systems (EEWS). However the limitations of this work is the training of the encoder put at the head of the CNN that simultaneously conducts feature extraction, down sampling, and learning about the event magnitude and location. There will be effort to estimate an accurate warning time for a particular site by including factors such as the nature of the event, the station data, and locations of the site. The author of [25] proposes a framework that integrates optical remote sensing (Sentinel-1) and GIS data to dynamically measure flood hazard and risk levels in the Trieste, Muggia Municipalities and Monfalcone. Explainable machine learning methods, particularly the Random Forest model, were used to generate flood hazard maps, achieving a high F1-score of approximately 0.99. Flood risk estimation combined a rule-based method for experience and vulnerability with dynamic flood hazard valuation for a comprehensive evaluation. However, the scarcity of annotated datasets is a severe issue for the domain regarding training and evaluation of ML models that are designed for the detection and monitoring of floods through remote sensing methods. Furthermore, the low resolution in terms of temporal acquisition frequency of satellite imagery is very challenging because it does not allow them to monitor floods in real time as they evolve.

III. PROBLEM STATEMENT

This research explores innovative data-driven solutions for improving disaster management by utilizing IoT-enabled predictive systems and real-time monitoring. It seeks to enhance disaster preparedness, minimize response times, and optimize resource allocation to mitigate the impact of natural and man-made crises effectively. Additionally, this study offers research solutions for the mentioned issues.

Specific Problem Definition:

The author in [26] focuses on developing an accurate flood susceptibility map for the Haraz watershed in Iran by means of an innovative modeling method known as DBPGA, optimized by the Genetic Algorithm by integration of the Deep Belief Network and the Back Propagation procedure. For this task, a database was designed using the ORAE technique containing ten training factors and 194 flood positions. The main issues are detailed below;

- ➢ The biggest challenge in disaster management is noisy input data or some irrelevant factors, which creates uncertainty. Smoothing out the noise and fitting by preprocessing may significantly give a boost to the correctness of the models for reliable predictions of disasters.

The author in [27] introduces an IoT-enabled Energy-Aware Metaheuristic Clustering with Routing Procedure for Real-Time Disaster Managing, EAMCR-RTDM. The EAMCR-RTDM approach is dedicated to optimizing node energy consumption with respect to the peculiarity of disaster-prone regions. For this reason, it uses a method called Yellow Saddle Goatfish-Based Clustering, YSGF-C, for the choice of cluster head and cluster organization. In addition, an Enhanced Cockroach Swarm Optimization algorithm for multi-hop routing, termed as ECSO-MHR, is proposed to discover the optimal routing paths. YSGF-C and ECSO-MHR both have employed fitness purposes based on varied input parameters to boost energy efficiency and prolong the network lifetime. The issues with this work are as follows:

- ➢ Here the limitations of these proposed methods were improved by the strategy of unequal clustering mechanism to resolve hot spot problems in the IoT networks.

The author in [28] demonstrates to classified earthquakes by their magnitude and impact, using global data from earthquakes between 1900 and 2021. Using historical patterns, numeric thresholds were determined to classify the magnitude, fatalities, injuries, and damage produced by earthquakes as low, medium, and high. Classification scheme after an earthquake incident can be used to estimate total loss by taking magnitude, number of people who have died and people that were injured, and money spent. The main issues are detailed below,

- ➢ The distribution and use of resources for emergency management are limited by the specific disaster class that determines the scope, allocation, and prioritization of resources. Effectively managing overlapping or unclassified disaster scenarios may become more challenging as a result.

The author in [29] explores SCALODEEP that is a deep learning-based detection basis of earthquakes by scalogram and skip connection, thereby extracting the higher order features out of three-component seismograms. The algorithm is based on the training data with North California, and test results show great accuracy while detecting seismic signals even for low magnitudes. Compared to the traditional model and new deep learning-based models recently developed, it outperforms with higher generalization. So, the work improves detection systems in the earthquake mechanism and sheds insight into making robust deep learning models for analysing seismicity. The challenges that come across those tactics are listed below,

- ➢ In this research, Data Overfitting and the reduced resolution of scalograms on SCALODEEP model where is challenging in terms of low efficiency and accuracy. To cope an overfitting, dropout with the rate of 0.2 is applied for making better generalization at trainability.

The author in [30] presents an IoT-assisted flood management system designed to collect data from both radar and environmental sensors to predict eventual floods. A 1D-CNN captures spatial patterns, but an M-LSTM handles the temporal dependences in multivariate time-series data, showing better results with an MSE of 0.018. The main issues are detailed below,

- ➢ Here the limitations of these methods such as CNN and LSTM models are previously less efficient, a deeper search for more innovative ML procedures, like Transformer models, may help to enhance the accurateness of predictions.

The author in [31] explores this further as an Internet of Things-related device used in improving emergency responses and disaster management using real-time sensor and IoT devices. It reviews potential benefits, challenges, and risks from deploying the IoES technology toward the public, in which its safety depends upon and what's at the root of public crisis and risk management. The issues with this work are as follows:

- A limitation of IoES implementation is the difficulty in meeting the different needs and preferences of various communities and stakeholders. Moreover, it can be complex and resource-intensive to ensure that these groups are effectively involved in decision-making processes.

Research solutions:

To address the problems addressed by cutting-edge methods, and to secure multi-user communication channels in the quantum network that concentrates on improving query efficiency, eavesdropping detection, and secure key exchange. To address the problems addressed by cutting-edge methods, and to predict the disaster in real-time monitoring using IoT sensors for enhancing the disaster management system. Initially we utilize the Denoising Autoencoder (DAE) will be used to handle the uncertainties in noisy input data and irrelevant factors. Thus, that noise removal and overfitting prevention during preprocessing will definitely improve the accuracy of the reliable disaster prediction models. This issue is to overcome the unequal clustering mechanism and solve the hot spot issues in IoT networks by using the Tuna-Swarm-Algorithm (TSA) for clustering. The optimal Resource Allocation minimizes the total risk: Fraction of unmet Demand, by distributing scarce emergency resources to clusters. Given the demand matrix, priority rating and the constraint on the available resource, the optimum is determined. It produces an allocation table with allocated resources in each cluster ranked according to priorities and demands. Effective overcoming of data overfitting, low resolution of proposed methods can embed Switchable Normalization into CNN architectures to select the best among different normalization methods: batches, layers, or instances of normalization and aid in achieving stable learning while minimizing the risk of overfitting. Armed with a high-quality time-frequency resolution technique, guarantees robust feature extraction and far better generalization. A hybrid model combines feature extraction using Mobile Net and the new generative Transformer model, which captures more complex dependencies in the data, thus yielding a better prediction accuracy. MobileNet, because of its light structure, offers rapid spatial feature learning, while the Transformer uses its attention mechanism to get temporal or contextual insights. As such, this fusion is the novelty called the MobileNet-Transformer generative model MN-TGM and streamlining the overall prediction considerably. This technique combines Fault Tree Analysis and DEMATEL (FTA-DEMATEL) for better decision-making and oversight in disaster management. FTA identifies potential failures in systems and their causes, while DEMATEL analyzes interdependence between factors that might influence disaster outcomes. The combination offers a novel approach to improving the quality of both preventive and responsive mechanisms in disaster management. Integration would allow for a more holistic view of analysis in terms of disaster risks and resource allocation.

IV. PROPOSED WORK

The proposed methodology utilizes IoT-enabled predictive analytics and real-time monitoring for disaster management. It involves synthetic data collection, denoising, clustering for resource allocation, and training to prevent overfitting. The hybrid prediction model, decision-making, ensures proper forecasting and effective disaster response. Fig 1 represents the overall architecture of proposed methodology. The proposed methods are as follows:

- **Synthetic data Collection using the IoT sensors**
- **Denoising the data**
- **Clustering and Resource Allocation**
- **Training to mitigate the overfitting**
- **Prediction**
- **Decision making**

A. Synthetic data collection using the IoT sensors

Synthetic data can be created using simulation models that simulate real environmental conditions to collect data about temperature, humidity, wind speed, and sensor readings from IoT devices. Such models can replicate a considerable number of scenarios such as various types of weather, sensors performance, and other features of the environment. This form of generating synthetic data proves helpful in testing and validating sensor networks and algorithms before these applications are used in real-world activities. This will ensure that it will work properly in most environmental settings by the IoT devices sensor- equipped to capture the data, which is then saved and processed and assessed in real time.

B. Denoising the Data

After the data collection we need to denoising the gathered data. The application of **Denoising Autoencoders (DAE)** offers a way for the removal of noise and irrelevant components that may otherwise affect the accuracy of such a prediction. DAE also filters uncertainties from the model caused by the quality of data input used. This mainly aims at improving model precision by removing all those extraneous elements that make this model prone to overfitting. This step focuses the system on the most relevant information, thereby allowing enhanced accuracy and performance in further analysis.

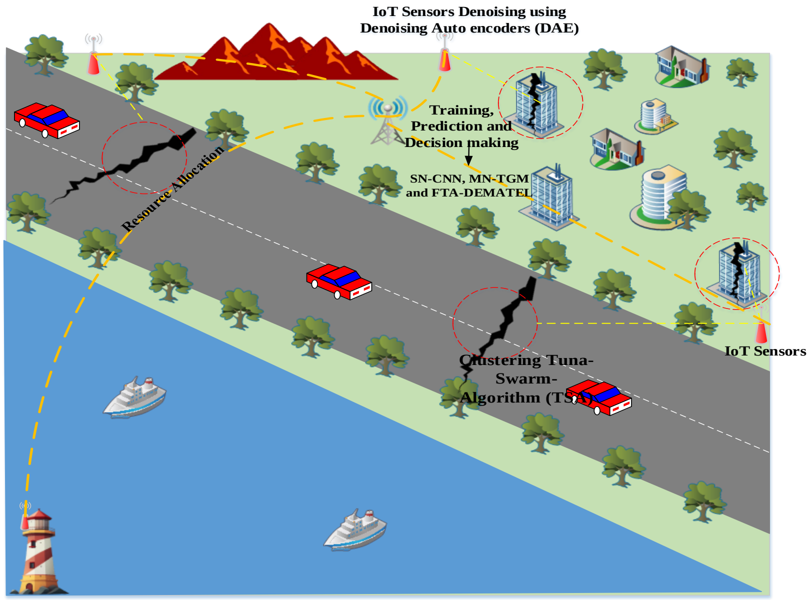

Fig 1 The overall architecture of proposed methodology

a. Denoising autocoders

Denoising Autoencoders (DAE) are neural networks that clean the input by removing noise. The testing starts with a corrupted input $\mathbb{x}_{noisy}$, which goes into the encoder and appears in the latent representation $\mathbb{y}$ after the processing logic being $\mathbb{y} = f_\theta(\mathbb{x}_{noisy})$. f_θ Is the Encoding function in the DAE is parameterized by θ. It maps noisy input $\mathbb{x}_{noisy}$ to latent space representation $\mathbb{y}$. The decoder produces the solution $\hat{\mathbb{x}}$ for the cleaning up and that is done in a way with the version: $\hat{\mathbb{x}} = g_\delta(\mathbb{y})$. The term g_δ is the decoding function in the DAE is parameterized by δ. It reconstructs $\hat{\mathbb{x}}$ from the latent representation $\mathbb{y}$. To optimize it, DAEs minimize a loss function ℓ that combines from distortion on one hand with spatial and spectral fidelity on the other. The term α is the weighting parameter regulates between spatial and spectral fidelity in the loss function.

$$\ell = \alpha . MSE + (1-\alpha). PSE \quad (1)$$

Mean square Error (MSE) provides accuracy with respect to the spatial domain, while the power spectrum error (PSE) ensures accuracy with respect to the frequency domain. M Is the number of samples and the $\mathbb{x}_i$ is the ith true data sample. The term H is the frequency bins in the power spectrum of the signal.

$$MSE = \frac{1}{M}\sum_{i=1}^{M}(\mathbb{x}_i - \hat{\mathbb{x}}_i)^2 \quad (2)$$

$$PSE = \frac{1}{H}\sum_{j=1}^{H}|\log T_{true}(j) - log T_{pred}(j)|^2 \quad (3)$$

Here, T_{true} and T_{pred} denote the power spectra of the true and predicted signals, respectively. This composite loss function allows DAEs to properly denoise the data whilst retaining various characteristics of the signals important for improving prediction accuracy and mitigating overfitting risks in predictive models.

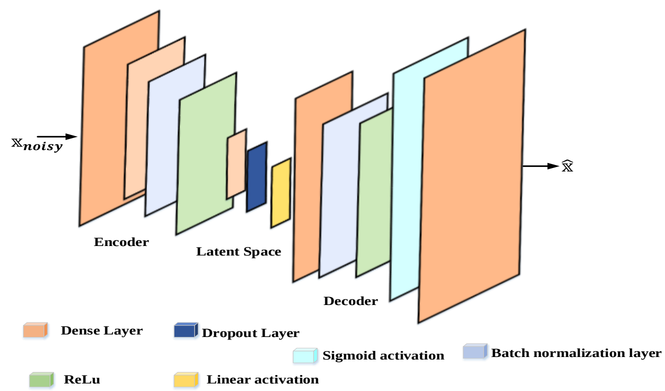

Fig 2 Denoising Autoencoder Architecture

Fig2 illustrates the architecture of denoising autoencoder. The Denoising Autoencoder (DAE) consists of several key components: an encoder, latent space and decoder. The encoder compresses the noisy input $\mathbb{x}_{noisy}$ into a latent representation using dense layers, batch normalization, ReLU activation, and dropout for regularization. The latent space contains the main features of the data in compact form. The decoder reconstructs the denoised output $\hat{\mathbb{x}}$, reversing the encoding process by passing through dense layers with linear and sigmoid activations. The architecture achieves robust noise reduction while maintaining input fidelity with a combination of activation and normalization techniques.

C. Clustering and Resource Allocation

Following denoising, the process is followed up by clustering. Here, **Tuna Swarm- Algorithm (TSA)** techniques are used on IoT data for group-cluster formation based on the demand and priority of patterns. Such a mechanism helps avoid hotspot issues and ensure proper distribution of resources. TSA lets the best-fit clusters in which minimum risks are guaranteed to obtain adequate resources for each cluster. Then, based on priorities and demands, an allocation table is prepared to deploy resources in real time, ensuring that the resources get distributed efficiently to mitigate the risks in disaster areas.

a. Tuna Swarm based clustering Algorithm

Tuna Swarm-Algorithm (TSA) is clustering-based optimization for IoT data management that simulates the schooling behaviour of tuna. The clustering is based on the demand and priority, ensuring effective resource allocation. The very first step of the algorithm states the initialization of some major parameters, such as the number of tunas $\mathfrak{n}$, the search space dimension $\mathbb{E}$, maximum number of iterations $\mathbb{I}$, and population of tunas so defined as

$$\mathcal{X}_{\mathrm{i}} = [\mathcal{X}_{\mathrm{i}1}, \mathcal{X}_{\mathrm{i}2}, \ldots, \mathcal{X}_{\mathrm{i}\mathbb{E}}] \tag{4}$$

Where, $\mathrm{i} = 1, 2, \ldots, \mathfrak{n}$. The actual candidate solution for the clustering process is given by the position of each tuna in the search space, expressed by $\mathcal{X}_{\mathrm{i}}$. To ascertain the relative quality of each position of the tuna, the fitness function $\mathfrak{F}(\mathcal{X}_{\mathrm{i}})$ is defined, one that aims to minimize risks $\mathcal{Q}(\mathcal{X}_{\mathrm{i}})$ while maximizing resource expenditure $\mathcal{E}(\mathcal{X}_{\mathrm{i}})$. The fitness function has this form:

$$\mathfrak{F}(\mathcal{X}_{\mathrm{i}}) = \mathfrak{v}_1 . \mathcal{Q}(\mathcal{X}_{\mathrm{i}}) + \mathfrak{v}_2 . \mathcal{E}(\mathcal{X}_{\mathrm{i}}) \tag{5}$$

By definition, $\mathcal{Q}(\mathcal{X}_{\mathrm{i}})$ indicates the current market risk for a given position, whereas $\mathcal{E}(\mathcal{X}_{\mathrm{i}})$ quantifies the compactness of the cluster. The weighted parameters $\mathfrak{v}_1$ and $\mathfrak{v}_2$ are obstacle weights employed in balancing the risk and objectivity with regard to the compactness of the cluster. Position updates for each tuna are according to swarming rules as determined by its local leader position $\mathcal{X}_{leader}$ and distance of the other tunas. The update rule is defined as:

$$\mathcal{X}_{\mathrm{i}}^{\mathbb{I}+1} = \mathcal{X}_{\mathrm{i}}^{\mathbb{I}} + \mathfrak{s} . \left(\mathcal{X}_{leader} - \mathcal{X}_{\mathrm{i}}^{\mathbb{I}}\right) \tag{6}$$

Where $\mathfrak{s}$ is a random number in the range of [0,1], which signifies an exploration parameter that defines the extent of randomness in the tuna movement.

To prevent hotspots, which occur when certain clusters become too concentrated, a penalty term $\mathcal{T}(\mathcal{X}_{\mathrm{i}})$ is introduced to ensure balanced clusters. The newly adjusted fitness function is:

$$\mathfrak{F}'^{(\mathcal{X}_{\mathrm{i}})} = \mathfrak{F}(\mathcal{X}_{\mathrm{i}}) + \lambda . \mathcal{T}(\mathcal{X}_{\mathrm{i}}) \tag{7}$$

Where λ is the penalty coefficient that regulates the effectiveness of the penalty term.

The cluster assignment of IoT data points is done by constructing the Euclidean distance between $\mathrm{e}_{\mathfrak{t}}$ and the respective cluster centre $\mathcal{X}_{\mathcal{E}\mathrm{i}}$. The distance is calculated as:

$$\mathrm{e}(\mathrm{e}_{\mathfrak{t}}, \mathcal{X}_{\mathcal{E}\mathrm{i}}) = \sum_{m=1}^{d} \left(\mathrm{e}_{\mathfrak{t}m} - \mathcal{X}_{\mathcal{E}\mathrm{i},m}\right)^2 \tag{8}$$

Since $\mathrm{e}_{\mathfrak{t}} = \left[\mathrm{e}_{\mathfrak{t}1,}\mathrm{e}_{\mathfrak{t}2,\ldots,}\mathrm{e}_{\mathfrak{t}d}\right]$signifies the IoT data point, and $\mathcal{X}_{\mathcal{E}\mathrm{i}} = [\mathcal{X}_{\mathcal{E}\mathrm{i}1}, \mathcal{X}_{\mathcal{E}\mathrm{i}2}, \ldots, \mathcal{X}_{\mathcal{E}\mathrm{i}d}]$ is the cluster centre. Each point represents a data point closest to the centre of a cluster according to this distance.

As the last, the algorithm iterates over clustering by updating the positions and re-evaluating fitness until convergence or the maximum number of iterations $\mathbb{I}$ is reached. This will ensure optimal clustering of IoT data effective resource management. The tuna swarm based clustering algorithm pseudocode was given below as pseudocode 1.

Pseudocode 1: Tuna Swarm based clustering Algorithm

1. Initialize the parameters:
   - $\mathfrak{n}$: Number of tunas (search agents)
   - $\mathbb{E}$: Search space dimension
   - $\mathbb{I}$: Maximum number of iterations
   - Population of tuna's eq.4
2. Initialize the candidate solutions (positions of tunas) in the search space.
3. For each tuna agent $\mathrm{i} = 1\ to\ \mathfrak{n}$:
   - Calculate the fitness function eq.5
   - Calculate the risk $\mathcal{Q}(\mathcal{X}_{\mathrm{i}})$ and resource expenditure $\mathcal{E}(\mathcal{X}_{\mathrm{i}})$
   - Set the initial cluster centre and evaluate fitness.
4. For each iteration($\mathrm{i} = 1\ to\ \mathbb{I}$):
   - Select the leader tuna $\mathcal{X}_{leader}$ with the best fitness value
   - Update the position of each tuna using the update rule using eq.6, where s is a random number in [0, 1]
   - Calculate the new fitness eq.7, introducing penalty term $\mathcal{T}(\mathcal{X}_{\mathrm{i}})$ to avoid hotspot issues.
   - Re-evaluate the fitness for each tuna.
   - Assign each IoT data point $\mathrm{e}_{\mathfrak{t}}$ to the nearest cluster centre $\mathcal{X}_{\mathcal{E}\mathrm{i}}$ based on the Euclidean distance using eq.8
   - Check for convergence (or max iterations reached). If converged, terminate; otherwise, continue iterating.
5. Return the optimal cluster centres as the result of the clustering process.

D. Training to Mitigate Overfitting

In order to enhance the model's performance and prevent overfitting, the system implements **Switchable Normalization within its SN-CNN** architecture. SN adaptively selects the best normalization strategy: Batch, Layer or Instance to stabilize learning in such a way that the model does not overfit the training data. This makes generalization better so that the model can handle unseen data even better. The advanced methods of time-frequency resolution ensure extracting the critical features with the maximum accuracy. This phase allows the model to process complex patterns and data more efficiently and increase predictive reliability. In the prediction phase, a hybrid model combining MobileNet for feature extraction and a Transformer-based generative model for capturing data dependencies is deployed.

a. Switchable Normalization within its SN-CNN

Switchable Normalization (SN) combines the principles and computations of Batch Normalization (BN), Layer Normalization (LN), and Instance Normalization (IN) to dynamically stabilize learning and reduce overfitting. SN adaptively learns the most suitable normalization by taking a weighted average, using their respective means and variances. The normalized output is computed as:

$$\widehat{\mathcal{H}} = \frac{\mathcal{H} - \mu_{\mathcal{MN}}}{\sqrt{\sigma^2_{\mathcal{MN}} + \omega}} \tag{9}$$

In this expression, $\mathcal{H}$ is the input feature map, $\mu_{\mathcal{MN}}$ is the weighted mean, $\sigma^2_{\mathcal{MN}}$ is the weighted variance, and ω is a small constant for numerical stability. The weighted mean $\mu_{\mathcal{MN}}$ and variance $\sigma^2_{\mathcal{MN}}$ is respectively given as:

$$\mu_{\mathcal{MN}} = \vartheta_{\mathrm{BN}}\mu_{\mathrm{BN}} + \vartheta_{\mathrm{LN}}\mu_{\mathrm{LN}} + \vartheta_{\mathrm{IN}}\mu_{\mathrm{IN}} \tag{10}$$

$$\sigma^2_{\mathcal{MN}} = \vartheta_{\mathrm{BN}}\sigma^2_{BN} + \vartheta_{\mathrm{BN}}\sigma^2_{LN} + \vartheta_{\mathrm{BN}}\sigma^2_{IN} \tag{11}$$

The means over batch-wise, layer-wise, and instance-wise dimensions are represented by the symbols μ_{BN},μ_{LN},μ_{IN}. In the same vein, the variances are represented by σ^2_{BN},σ^2_{LN},σ^2_{IN}. The weights ϑ_{BN}, ϑ_{LN} , ϑ_{IN} are learnable parameters that meet the following conditions:

$$\vartheta_{\mathrm{BN}} + \vartheta_{\mathrm{LN}} + \vartheta_{\mathrm{IN}} = 1 \tag{12}$$

SN enhances the generalization process by calming the distribution at every dimension and adapting itself to the variety in data distributions. After normalization, the features are passed through a convolution operation:

$$\mathscr{p} = \mathbb{w} * \widehat{\mathcal{H}} + b \tag{13}$$

Where, $\mathbb{w}$ is the convolution filter, b is the bias, and $*$ denotes the convolution operation. Non-linearity is introduced using the ReLU activation function:

$$\rho' = max(0, \rho) \tag{14}$$

Dropout ($\mathscr{h}$) is another way of tackling overfitting:

$$\mathscr{h} = \mathscr{p}'.r \tag{15}$$

Where, r random mask of retention probability ρ. Finally, features are classified via the softmax function ($\mathfrak{Y}$):

$$\mathfrak{Y} = \frac{exp(\mathcal{V}_{\mathscr{f}}.\mathfrak{S} + \mathcal{B}_{\mathscr{f}})}{\sum_{\mathcal{J}} exp\left(\mathcal{V}_{\mathscr{f}}^{(\mathcal{J})}.\mathfrak{S} + \mathcal{B}_{\mathscr{f}}^{(\mathcal{J})}\right)} \tag{16}$$

Where, $\mathcal{V}_{\mathscr{f}}$ and $\mathcal{B}_{\mathscr{f}}$ are the weights and biases of the fully connected layer, respectively and $\mathcal{J}$ indexes the output classes. The strong normalizing process guarantees a stable model and limits overfitting, giving the model better generalization for variable data conditions. Fig 3 shows the graphical representation of switchable normalization.

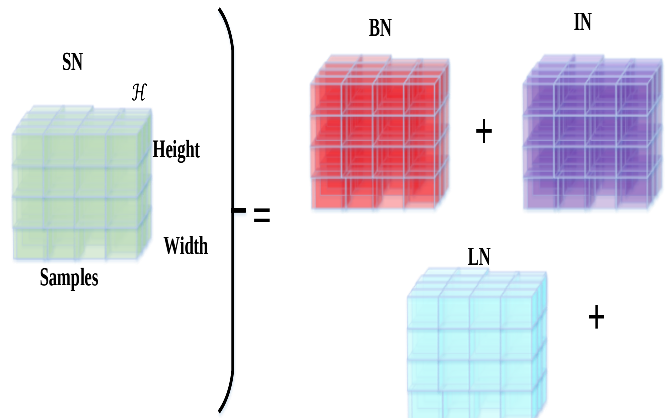

Fig 3 Graphical representation of Switchable Normalization

E. Prediction

In this phase, the spatial features are processed efficiently by the MobileNet component, and the Transformer model uses its attention mechanism to understand temporal and contextual relationships. It will develop the final model approach, called a **MobileNet-Transformer Generative Model (MN-TGM)**. Such an enhancement of prediction by the system will provide the disaster management teams with their timeliness and accuracy so as to allow taking proper action before disaster occurrence. Finally, it is coupled with Fault Tree Analysis and Decision-Making Trial and Evaluation Laboratory in supporting strategic decision making and resource management. FTA outlines reasons for potential system failures with their root cause; DEMATEL quantifies interrelation between several factors that affect outcomes of disasters.

a. MobileNet

The traditional two-step procedure for extracting spatial features from input otherwise is depthwise separable convolution, which is adopted by MobileNet to achieve efficient implementation with much reduced computational complexity. Breaking standard convolution operations into spatial and channel-spaced computations makes this method more appropriate for resource-limited environments. The first one is depthwise convolution, which individually applies one filter per input channel. This approach captures spatial patterns for all the channels while holding low parameter counts.

A mathematical formulation for depthwise convolution ($A_{depth}(k,l)$) is given as:

$$A_{depth}(k,l) = \sum_{p=1}^{P} k_p * l_p \tag{17}$$

Where, k_p represents the p^{th}channel of the input feature map, l_p is the depthwise kernel corresponding to k_p, and $*$ denotes the convolution operation.

The second step is the pointwise convolution $A_{point}(A_{depth}, \mathrm{E})$, where the outputs from the depthwise convolution are recombined by way of 1×1 convolutions. This step allows for interaction between channels for models to learn complex patterns. It can be expressed as

$$A_{point}(A_{depth}, \mathrm{E}) = \sum_{q=1}^{Q} E_q . A_{depth,q} \tag{18}$$

Where, E_q is the 1×1 conv kernel for the q^{th}output channel, and q is the number of output channels. Putting these two together gives the overall process of depthwise separable convolution (A_{sep}) with eq.19

$$A_{sep} = A_{point}(A_{depth}) \tag{19}$$

This approach is efficient as it dramatically cuts down on the computational cost compared to standard convolutions. For standard convolutions, the cost $Cost_{std}$ is simply

$$Cost_{std} = D_l^2 . P . Q . D_g^2 \tag{20}$$

Where D_l^2 the spatial size of the kernel, P is the number of input channels, Q is the number of output channels, and D_g^2 represents the spatial dimensions of the feature map. In contrast, the cost of the depthwise separable convolution $Cost_{sep}$ is given by,

$$Cost_{sep} = D_l^2 . P . D_g^2 + P . Q . D_g^2 \tag{21}$$

The computational savings can be quantified as

$$\frac{Cost_{sep}}{Cost_{std}} = \frac{1}{P} + \frac{1}{D_g^2} \tag{22}$$

It proves that the operations are burrowing faster. This efficiency makes MobileNet very suitable for real-time applications, for example, disaster forecasting, where speed of processing and resourcing in the current trend are very critical. Supported by depthwise separable convolutions, MobileNet offers a balance between speed and precision of efficient feature extraction and, therefore, represents a strong candidate for time-sensitive scenarios that entail rapid and accurate spatial analysis.

b. Transformer Generative Model

A self-attention mechanism by the transformers operates on an input sequence after recognizing inter-feature relationships and dependencies concerning time. This approach proves immensely useful with disaster prediction, wherein spatial features from MobileNet are merged with temporal features. The self-attention process serves to provide different importance to each feature regarding the relation to every other feature within a sequence, directing the model towards the robust points in the data for accurate predictions. The self-attention operation calculates the impact of one feature on another within the input sequence such that outputs are generated by weighting features by their contextual significance.

The self-attention mechanism is mathematically defined as:

$$Attention\ (\mathbb{H}, \mathcal{U}, \mathcal{Z}) = softmax\left(\frac{\mathbb{H}\mathcal{U}^T}{\sqrt{e_{\mathcal{U}}}}\right)\mathcal{Z} \tag{23}$$

Here, $\mathbb{H}$ (query matrix), $\mathcal{U}$ (key matrix), and $\mathcal{Z}$ (value matrix) and T represents the transpose that are derived from the input features. The term $\frac{\mathbb{H}\mathcal{U}^T}{\sqrt{e_{\mathcal{U}}}}$ computes the similarity between the query and key vectors, scaled by $e_{\mathcal{U}}$ to ensure numerical stability. The resulting values are passed through the softmax function to obtain probabilities that assign a certain level of importance to the features. The weighted output is then calculated through multiplication of the weights with the value matrix $\mathcal{Z}$.

Weights assigned to the features:

$$Weights = softmax\left(\frac{\mathbb{H}\mathcal{U}^T}{\sqrt{e_{\mathcal{U}}}}\right) \tag{24}$$

The softmax function is defined as

$$Softmax\ (c_{\mathfrak{y}}) = \frac{e^{c_{\mathfrak{y}}}}{\sum_{\mathfrak{m}=1}^{\mathfrak{o}} e^{c_{\mathfrak{y}}}} \tag{25}$$

The softmax function indicates that $c_{\mathfrak{y}}$ corresponds to the raw score (otherwise known as the logit) for class $\mathfrak{y}$ in a classification problem, before it is converted into a probability. $\mathfrak{m}$ is the term that represents the adding the overall possible classes.

Any set of weights will fit on a unit-interval probability distribution and represent the relative importance of each feature.

The multi-head mechanism extends this process by enabling the model to focus on different parts of the input data simultaneously. That is expressed as:

$$MHA(\mathbb{H}, \mathcal{U}, \mathcal{Z}) = concat(head_1, head_2, \ldots, head_{\hbar})\mathcal{Z} \quad (26)$$

Here, $head_{\hbar}$ indicates the attention output for the $\hbar$ attention head, while $W^{\mathfrak{F}}$ is the learned weight matrix which combines the results from all attention heads.

After computing the self-attention outputs, it is passed through a feed-forward neural network that composes it into a more refined format. The feedforward operation is expressed as:

$$FFN(c) = ReLu(cW_1 + \mathcal{B}_1)\, W_2 + \mathcal{B}_2 \quad (27)$$

Weights W_1 and W_2 are paralleled matrices, whereas, in the case of biases $\mathcal{B}_1$ and $\mathcal{B}_2$, they are the appertaining biasing terms.

In disaster prediction, through these computations, the model Transformer can dynamically shift its attention among various spatial and temporal features, allowing it to generate highly contextual outputs. By combining features extracted from MobileNet with temporal analysis capabilities of the Transformer, critical patterns and relationships in disaster data can be identified for better timing and accuracy for predictions. This would make the Transformer model for capturing the complex settings sufficient enough for putting into effect disaster management.

F. Decision Making

The Strategic Plan focuses upon the **Fault Tree Analysis as well as Decision-Making Trial and Evaluation Laboratory methods (FTA-DEMATEL)** for developing decisions and resource management. It evaluates the possible system failure and its underlying causes according to FTA. As on the other hand, it analyses the interrelation between different factors that influence the disasters. The integrated method developed by FTA and DEMATEL reflects a detailed view of disaster risk for superior capability development at decision-making. This gives a robust framework for preventive measures and real-time response strategies that enable disaster management teams to address the upcoming crises in time.

a. Fault Tree Analysis

The Fault Tree Analysis (FTA) is a deductive, top-down method used to analyse the failure of systems by modelling the relationship that may exist between a cause or causes of failure and a primary undesired event, known as the top event. The top event-for instance, a system failure or disaster disruption-is treated alone through logical gates (e.g., an OR gate and an AND gate), which models its sub-events. The probability of a top event ($\wp_{top}$)

$$\wp_{top} = 1 - \prod_{q=1}^{x}(1 - \wp_q) \quad (28)$$

Where, $\wp_q$ represents the probability of basic event q. On the other hand, for an AND gate, the equation is

$$\wp_{top} = \prod_{q=1}^{x} \wp_q \quad (29)$$

Probabilities are assigned to these basic events based on historical data or expert perceptions. When discussing various hazards, where disruption from a facility due to an earthquake ($\wp_{ek}$) or a flood ($\wp_{fd}$), then the combined probability for failure ($\wp_{failure}$) is

$$\wp_{failure} = 1 - (1 - \wp_{ek})(1 - \wp_{fd}) \quad (30)$$

Simply put, in an AND gate where both hazards must operate concurrently to result in failure, and the probability is given by

$$\wp_{failure} = \wp_{ek} \cdot \wp_{fd} \quad (31)$$

Moreover, in a complex event that required conditional probabilities such as disruption due to both facility Vulnerability ($\wp_{vb}$) and hazard occurrence($\wp_{Ho}$), the probability of event $\wp_{event}$ expression can be

$$\wp_{event} = \wp_{vb} \cdot \wp_{Ho} \quad (32)$$

This calculation ultimately helps in disaster management by locating fragilities and deciding what to prioritize for mitigation.

b. Decision-Making Trial and Evaluation Laboratory methods

The acronym DEMATEL stands for Decision-Making Trial and Evaluation Laboratory-a methodology that could analyse and visualize the cause and effect relationships among the factors of a complex system through distinguishing causes or driving group from effects or dependent group. DEMATEL employs a direct-influence matrix $\hbar_{jr}$, where $\hbar_{jr}$ specifies the influence of factor j on factor r. This matrix is normalized as

$$W_{jr} = \frac{\hbar_{jr}}{\max(\sum_r \hbar_{jr}, \sum_r \hbar_{jr})} \quad (33)$$

To provide comparative values for the factors involved. The total influence matrix $\mathbb{C}$ takes into account direct and indirect influences, mathematically

$$\mathbb{C} = (\mathcal{I} - W)^{-1} \quad (34)$$

Where $\mathcal{J}$ denotes an identity matrix. Prominence analysis is defined as $(+RW + CM)$; where RW: row-sum and CM: column-sum matrices of solution $\mathbb{C}$ determining the total significance of a factor. The relationship $(-RW - CM)$will determine whether that factor mainly acts as a cause $(+)$ or as an effect$(-)$.

For example, if factor $\mathfrak{P}$ affects factors $\mathfrak{Q}$ and $\mathfrak{R}$ with the normalized weights $W_{\mathfrak{P}\mathfrak{Q}}$ and $W_{\mathfrak{P}\mathfrak{R}}$, respectively, the Total Influence Matrix for $T\mathfrak{P}$ is updated with

$$T\mathfrak{P}_{\mathfrak{P}} = \left(\mathscr{b} - W_{\mathfrak{P}}\right)^{-1} - \mathscr{b} \tag{35}$$

Similarly, the relative importance of a factor in the context of disaster management like preparedness $\mathfrak{P}r$ and infrastructure $\mathscr{b}$ could be represented with relationships such as

$$T\mathfrak{P}_{\mathfrak{P}r\to\mathscr{b}} \quad \text{or} \quad T\mathfrak{P}_{\mathscr{b}\to\mathfrak{P}r} \tag{36}$$

Thereby prioritizing actions such computations guide decision-makers toward determining crucial interdependencies and developing viable plans of action. Fig 4 represents the protocol for the DEMATEL method.

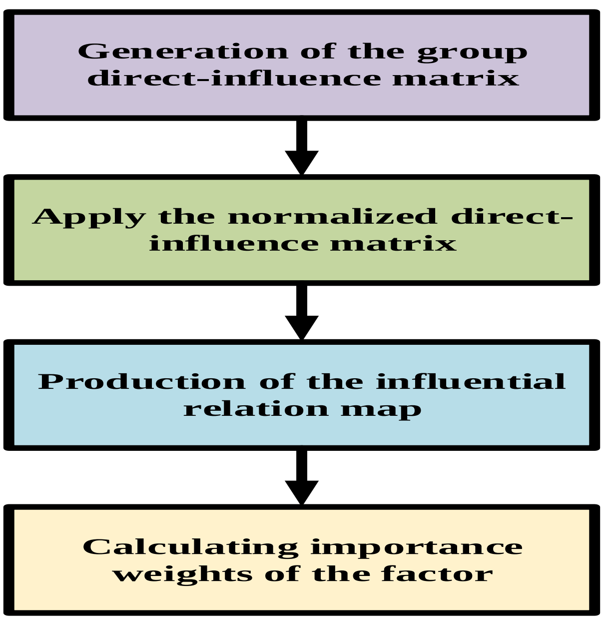

Fig 4 the protocol for DEMATEL method

V. EXPERIMENTAL RESULTS

This section presents the suggested research plans empirical and performance evaluations. This part is divided into three subsections: research summary, comparative analysis, and simulation study.

A. Simulation Study

To implement the suggested research approaches, ns-3.30.1 with python is employed and the Ubuntu 18.04 single primary operating system is utilized. Table 1 and 2 signifies the system specifications and the simulation parameter. Fig 5 represents simulation environment of this research.

TABLE 1
System specifications

Software Specifications	OS	Ubuntu 18.04
	Network Simulator	ns-3.30.1 with python
Hardware Specifications	RAM	4GB
	Hard Disk	500GB

TABLE 2
SIMULATION PARAMETER

Parameters		**Descriptions**
Network Parameters	IoT Nodes	4
	Lighthouse	1
	Base station	1
	vehicles	10

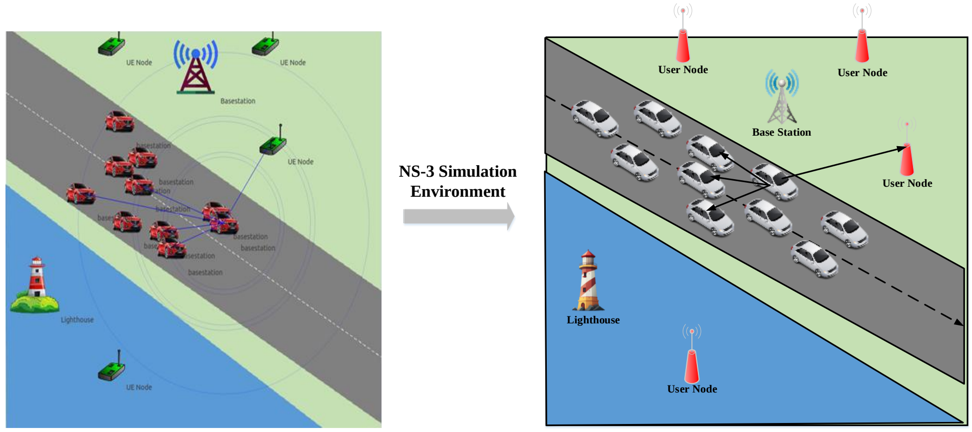

Fig 5 simulation environment of this research

B. Comparative Analysis

Here this segment we compare the suggested methods with existing methods such as “cluster based routing protocol for information centric wireless sensor network (CBR-ICWSN) [18]”, improved deep convolutional neutral network (IDCNN)[32], rain forest-decision tree (TF-DT) [10], k-means clustering (K-MC) [28]. The performance metrics were details described one by one.

a. Number of IoT nodes vs. Latency (ms)

The latency becomes a vital aspect to consider in IoT-enabled disaster management systems because of the number of IoT nodes in the network. Latencies tend to increase with the scaling up of the number of such nodes, primarily because of communication overhead, processing delays, and congestion in the network. The above can be represented as:

$$L = L_0 + k.Mo^{\Psi} \quad (37)$$

Where,

L is represents as total latency in milliseconds.

L_0 is the base latency

Mo is the number of IoT nodes in the network.

k is the proportionality constant dependent on the network capacity

Ψ is a non-linear scaling effect of a node is reflected in the exponent above.

TABLE 3
Numerical outcomes of Latency (ms)

(x-axis) – Number of IoT nodes	Latency (ms)- (y-axis)		
	CBR-ICWSN	KMC	Proposed
1	16	14	10
2	20	18	12
3	25	22	16
4	35	30	20

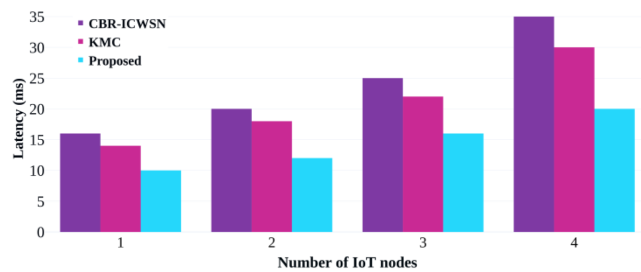

Fig 6 Number of IoT nodes vs. Latency (ms)

The Fig 6 and table 3 shows the latency values expressed in milliseconds for different algorithms such as CBR-ICWSN, KMC, and Proposed for different numbers of IoT nodes. Latency increased for all three methods as the number of IoT nodes increased. The latency of CBR-ICWSN starts from 16 ms and increases moderately until it reaches 35 ms as the node count is raised to four. The KMC algorithm starts -from 14 ms for the first node to 30 ms for the fourth node. The Proposed method achieves the lowest latency among all the schemes for any number of nodes, starting from 10 ms with one node to a maximum of 20 ms for four nodes. Hence, the proposed method has performed much better under a greater number of IoT nodes than CBR-ICWSN and KMC methods, which experience far more noticeability in terms of increasing latency as the number of nodes increases.

b. Number of IoT nodes vs. Overfitting Rate (%)

IoT nodes deployed in predictive analytics towards disaster management can complicate model training and increase risks of overfitting. Overfitting means that this process resulted in a model that has internalized patterns related to the particular training data instead of generalizations for unseen data. Such relation is mathematically expressed as:

$$O_f = \Psi . N^2 + \beta \tag{38}$$

O_f : Overfitting rate (%)

Ψ : Proportionality constant reflecting the sensitivity of overfitting to the number of nodes

β : Baseline overfitting rate without IoT nodes

TABLE 4
Numerical outcomes of Overfitting Rate (%)

(x-axis) – Number of IoT nodes	Overfitting Rate (%)- (y-axis)		
	IDCNN	KMC	Proposed
1	15	12	10
2	20	18	12
3	35	30	20
4	65	45	35

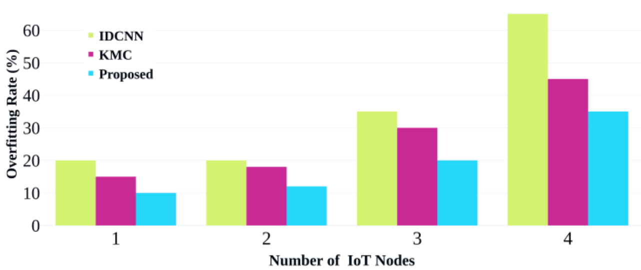

Fig 7 Number of IoT nodes vs. Overfitting Rate (%)

The comparison of the overfitting rates across different IoT nodes shows in Fig 7 and table 4. while IDCNN method exhibits by far the most significant escalation in overfitting rate beginning at 15% for 1 node and soaring up to 65% for 4 nodes, KMC, although rising, has this increase at a slower pace, going from 12% for 1 node to 45% for 4 nodes. The Proposed method shows, however, that overfitting rates increase much more slowly from an initial 10% for 1 node up to 35% for 4 nodes. This indicates the better generalization power of the proposed method with increasing number of IoT nodes, hence less proneness to overfitting compared to IDCNN and KMC. The overall impression would therefore be that the proposed method is more robust in dealing with up-scaled IoT node configurations.

c. Number of IoT nodes vs. Prediction Accuracy (%)

The relationship between IoT node count N, and prediction accuracy P, in disaster management could be described as:

$$P = P_{max}(1 - e^{-uN}) \tag{39}$$

P : Prediction accuracy

P_{max} : Maximum prediction accuracy that can be attained

u : Sensitivity factor; how quickly the prediction accuracy improves with the increase in the number of nodes.

As N increases, P asymptotically approaches P_{max}, indicating diminishing returns after a certain density of IoT nodes.

TABLE 5
Numerical outcomes of Prediction Accuracy (%)

(x-axis) – Number of IoT nodes	Prediction Accuracy (%)- (y-axis)		
	RF-DT	KMC	Proposed
1	20	30	40
2	30	50	70
3	40	60	80
4	60	70	90

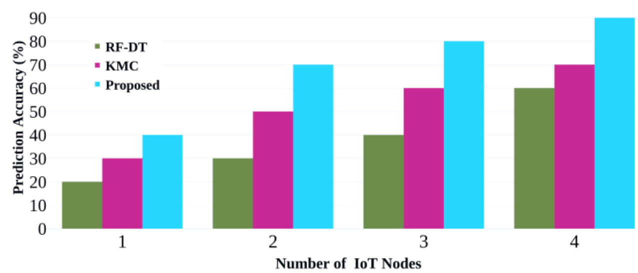

Fig 8 Number of IoT nodes vs. Prediction Accuracy (%)

Table 5 and Fig 8 show the prediction accuracy percentage achieved by RF-DT, KMC, and the proposed method, which cover various numbers of IoT nodes. The results point to a general increase in prediction accuracy with the growing number of nodes. For a single IoT node, the Proposed method reached an accuracy level of 40%, ahead of RF-DT, which had an accuracy of 20%, followed by KMC with 30% accuracy. With two nodes, the proposed method reached 70%, which was again very much ahead of RF-DT at 30% and KMC at 50%. For three nodes, the proposed method now reaches 80%, while RF-DT and KMC are behind at 40% and 60%, respectively. Finally, with four nodes, the proposed method projected a peak accuracy of 90%, while KMC and RF-DT lag at 70% and 60%, respectively. All of this demonstrates the dominating performance of the proposed method with respect to prediction, especially when the system is scaled up.

d. Number of Epochs vs. Model Accuracy (%)

In the context of IoT-driven predictive analytics for disaster management, epochs stand for the number of iterations the model undergoes during training. The more epochs trained, the higher becomes the model accuracy, as long as sufficient epochs do not overtrain and produce overfitting. The relation between epochs and accuracy is generally represented in the following form:

$$MA(E) = \frac{1}{1 + e^{-(wE+b)}} \tag{40}$$

$MA(E)$: Model accuracy after E epochs

w : represents the weight (learning rate) influencing the rate of accuracy improvement.

b : is the bias term determining the initial accuracy level.

This sigmoid curve explains that there is an increasing accuracy with higher numbers of epochs until such a time as the function flattens out.

TABLE 6
Numerical outcomes of Model Accuracy (%)

(x-axis) – Number of IoT nodes	Model Accuracy (%)- (y-axis)		
	RF-DT	KMC	Proposed
1	15	20	25
2	30	40	50
3	50	60	75
4	70	90	95

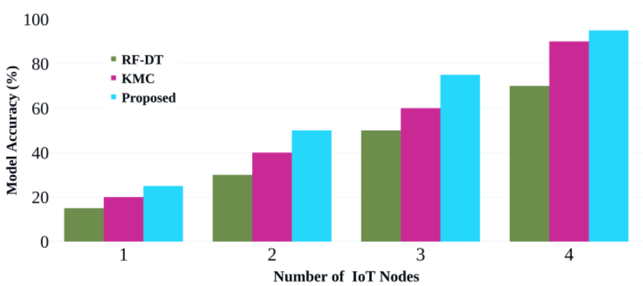

Fig 9 Number of Epochs vs. Model Accuracy (%)

The numerical values in Table 6 and Fig 9 reflect a comparative analysis of model accuracies of the RF-DT, KMC, and Proposed approaches with respect to the number of IoT nodes. With an increase in the number of nodes, a steady increase in the model accuracy was observed in all models, with the proposed approach outperforming all others in every instance. To provide an example, with 1 node, RF-DT reported accuracy at 15%, KMC at 20%, and the proposed method at 25%. At 4 nodes, the accuracy steadily rises to 70% in RF-DT, 90% in KMC, 95% in the proposed approach. This trend shows that there is a constant increase in the accuracy level afforded by the proposed method, which shines brighter at higher node levels where efficient handling of IoT data becomes crucial. Therefore, the proposed approach can be seen as a more dependable choice for predictive analytics with IoT support.

e. Number of Clusters vs. Energy Efficiency (%)

The number of clusters impacts energy efficiency in IoT systems by determining the way data is partitioned and processed. A higher number of clusters will reduce energy consumption through task distribution but may exhibit higher communication overhead. This relationship can be expressed as:

$$EE = \frac{1}{1+\propto .c} \tag{41}$$

- EE which is energy efficiency (%);
- c is the number of clusters;
- $\propto$ is a constant which represents the overhead cost per cluster.

The equation suggests that energy efficiency improves as the number of clusters increases to some point and thereafter decreases with the increase of communication overheads.

TABLE 7
Numerical outcomes of Energy Efficiency (%)

(x-axis) – Number of Clusters	Energy Efficiency (%)- (y-axis)		
	CBR-ICWSN	KMC	Proposed
1	15	25	30
2	30	45	50
3	50	65	70
4	75	85	92

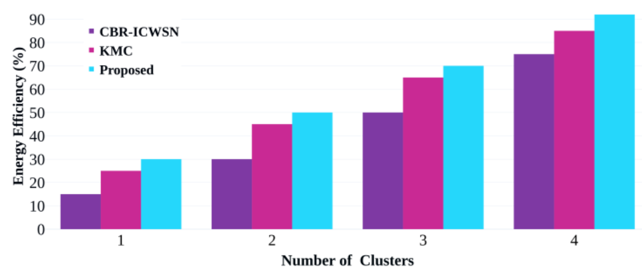

Fig 10 Number of Clusters vs. Energy Efficiency (%)

The energy efficiency comparison among the methods CBR-ICWSN, KMC, and Proposed shows a trend of improvement with an increase in the number of clusters in table 7 and Fig 10. At one cluster, the energy efficiency remains low in CBR-ICWSN at 15%; KMC, at 25%; while the Proposed method achieves 30%. There is a huge leap in efficiency at two clusters with CBR-ICWSN, KMC, and the proposed approaches increasing to 30%, 45%, and 50%, respectively. This trend persists with three clusters, where efficiency rises to 50%, 65%, and 70%. With four clusters, the proposed method is more efficient, being 92% as opposed to KMC, which is 85%, and CBR-ICWSN, which is 75%. This means that the proposed method has a better energy core always compared to both KMC and CBR-ICWSN in all cluster numbers, especially with an increase in the number of clusters.

C. Research summary

The IoT devices use real-time data and predictive analytics in a better way of managing disaster forecasts and responses. Meanwhile, other methods still face challenges such as the denoising effect, hotspots, improper allocation of resources, overfitted data, and complexity of the embedded decision-making processes. To address these issues, a predictive analytics framework for IoT-based disaster management is established.The framework consists of the creation of a network of IoT nodes, a lighthouse, a base station, and vehicles that generate the synthetic data. A Dual Denoising Autoencoder has been used to enhance the quality of data, and the Tuna-Swarm Algorithm for optimizing the clustering and resource allocation process. Switchable Normalization approaches used in a custom SN-CNN mitigated the overfitting of models. Mobile Net-Transformer Generative Model (MN-TGM) was used for prediction. Finally, Fault Tree Analysis and the Decision-Making Trial and Evaluation Laboratory (FTA-DEMATEL) enable decision-making. Ns3.30.1 simulation was used to validate the performance metrics like latency (ms), the prediction accuracy (%), model accuracy (%), and the effective energy efficiency (%). The framework will assure timely interventions, exhaustive utilization of resources, and effective disaster management. The performance metrics were elaborately in Fig 6-10 and Table 3- 7.

VI. CONCLUSION

The predictive analytics system for IoT disaster management solves critical issues of existing methods regarding denoising effects, inefficiencies in resource allocation, data overfitting, and decision-making complexities. A robust network consisting of IoT nodes, a lighthouse, a base station, and vehicles, along with advanced techniques like Denoising Autoencoder, Tuna-Swarm Algorithm, and Switchable Normalization with SN-CNN, has guaranteed better data quality, optimized resource utilization, and increased model performance. The mobile net-transformer generative model MN-TGM enables accurate and timely disaster predictions, whereas FTA-DEMATEL streamlines decision-making processes. Simulations conducted in Ns3.30.1 validate the system's efficiency across important performance parameters, such as latency, prediction accuracy, model accuracy, and energy efficiency. This framework thereby gives an all-encompassing solution for IoT-enabled disaster management, ensuring timely interventions, efficient resource utilization, and reliable decision-making processes, which ultimately reduce risks due to disasters and the impact of those disasters.

REFERENCES

1. Li, C., Wang, J., Wang, S., & Zhang, Y. (2024). A review of IoT applications in healthcare. Neurocomputing, 565, 127017.
2. Yu, T., & Wang, X. (2022). Real-time data analytics in internet of things systems. In Handbook of real-time computing (pp. 541-568). Singapore: Springer Nature Singapore.
3. Sicari, C., Catalfamo, A., Carnevale, L., Galletta, A., Balouek-Thomert, D., Parashar, M., & Villari, M. (2023, July). Tema: Event driven serverless workflows platform for natural disaster management. In 2023 IEEE Symposium on Computers and Communications (ISCC) (pp. 1-6). IEEE.
4. Biabani, Morteza, Hossein Fotouhi, and Nasser Yazdani. "An energy-efficient evolutionary clustering technique for disaster management in IoT networks." Sensors 20, no. 9 (2020): 2647.
5. Liu, J., Chen, Y., & Chen, Y. (2021). Emergency and disaster management-crowd evacuation research. Journal of Industrial Information Integration, 21, 100191.
6. K. Sharma, D. Anand, M. Sabharwal, P. K. Tiwari, O. Cheikhrouhou et al., “A disaster management framework using internet of things-based interconnected devices,” Mathematical Problems in Engineering, vol. 2021, no. 9916440, 2021.
7. Khanh, Q.V.; Hoai, N.V.; Manh, L.D.; Le, A.N.; Jeon, G. Wireless communication technologies for IoT in 5G: Vision, applications, and challenges. Wirel. Commun. Mob. Comput. 2022, 2022, 1–12.
8. Hsiao WT, Barnes EA, Maloney ED, Tulich SN, Dias J, Kiladis GN (2022) Role of the Tropics in State-Dependent Improvements of US West Coast NOAA Unified Forecast System Precipitation Forecasts. Geophys Res Lett 49(5).
9. Cicek, D., & Kantarci, B. (2023). Use of mobile crowdsensing in disaster management: A systematic review, challenges, and open issues. Sensors, 23(3), 1699.
10. Aljohani, F. H., Abi Sen, A. A., Ramazan, M. S., Alzahrani, B., & Bahbouh, N. M. (2023). A smart framework for managing natural disasters based on the iot and ml. Applied Sciences, 13(6), 3888.
11. Islam, M. A., Rashid, S. I., Hossain, N. U. I., Fleming, R., & Sokolov, A. (2023). An integrated convolutional neural network and sorting algorithm for image classification for efficient flood disaster management. Decision Analytics Journal, 7, 100225.
12. Khan, S. M., Shafi, I., Butt, W. H., Diez, I. D. L. T., Flores, M. A. L., Galán, J. C., & Ashraf, I. (2023). A systematic review of disaster management systems: approaches, challenges, and future directions. Land, 12(8), 1514.
13. Lyu, J., Zhou, S., Liu, J., & Jiang, B. (2023). Intelligent-Technology-Empowered Active Emergency Command Strategy for Urban Hazardous Chemical Disaster Management. Sustainability, 15(19), 14369.
14. Brar, P. S., Shah, B., Singh, J., Ali, F., & Kwak, D. (2022). Using modified technology acceptance model to evaluate the adoption of a proposed IoT-based indoor disaster management software tool by rescue workers. Sensors, 22(5), 1866.
15. Rangra, A., & Sehgal, V. K. (2022). Natural disasters management using social internet of things.

Multimedia Tools and Applications, 81(24), 34447-34461.

16. Jalaludin, S. (2024). Evaluation of Human Resource Performance at the Regional Disaster Management Agency in West Nusa Tenggara Province. Society, 12(2), 365-380.

17. Anjum, M. J., Anees, T., Tariq, F., Shaheen, M., Amjad, S., Iftikhar, F., & Ahmad, F. (2023). Space-Air-Ground Integrated Network for Disaster Management: Systematic Literature Review. Applied Computational Intelligence and Soft Computing, 2023(1), 6037882.

18. T. Vaiyapuri, V. S. Parvathy, V. Manikandan, N. Krishnaraj, D. Gupta et al., “A novel hybrid optimization for cluster-based routing protocol in information-centric wireless sensor networks for IoT based mobile edge computing,” Wireless Personal Communications, pp. 1–24, 2021.

19. S. Rajendran, O. I. Khalaf, Y. Alotaibi and S. Alghamdi, “MapReduce-Based big data classification model using feature subset selection and hyperparameter tuned deep belief network,” Scientific Reports, vol. 11, no. 24138, 2021.

20. Dai, W., Tang, Y., Zhang, Z., & Cai, Z. (2021). Ensemble learning technology for coastal flood forecasting in internet-of-things-enabled smart city. International Journal of Computational Intelligence Systems, 14(1), 166.

21. Prakash, C., Barthwal, A., & Acharya, D. (2022). FLOODWALL: a real-time flash flood monitoring and forecasting system using IoT. IEEE Sensors Journal, 23(1), 787-799.

22. Abid, S. K., Sulaiman, N., Chan, S. W., Nazir, U., Abid, M., Han, H., ... & Vega-Muñoz, A. (2021). Toward an integrated disaster management approach: how artificial intelligence can boost disaster management. Sustainability, 13(22), 12560.

23. Aamir, M., Ali, T., Irfan, M., Shaf, A., Azam, M. Z., Glowacz, A., ... & Rahman, S. (2021). Natural disasters intensity analysis and classification based on multispectral images using multi-layered deep convolutional neural network. Sensors, 21(8), 2648.

24. Abdalzaher, M. S., Soliman, M. S., El-Hady, S. M., Benslimane, A., & Elwekeil, M. (2021). A deep learning model for earthquake parameters observation in IoT system-based earthquake early warning. IEEE Internet of Things Journal, 9(11), 8412-8424.

25. Antzoulatos, G., Kouloglou, I. O., Bakratsas, M., Moumtzidou, A., Gialampoukidis, I., Karakostas, A., ... & Kompatsiaris, I. (2022). Flood hazard and risk mapping by applying an explainable machine learning framework using satellite imagery and GIS data. Sustainability, 14(6), 3251.

26. Shahabi, H., Shirzadi, A., Ronoud, S., Asadi, S., Pham, B. T., Mansouripour, F., ... & Bui, D. T. (2021). Flash flood susceptibility mapping using a novel deep learning model based on deep belief network, back propagation and genetic algorithm. Geoscience Frontiers, 12(3), 101100.

27. Santhanaraj, R. K., Rajendran, S., Romero, C. A. T., & Murugaraj, S. S. (2023). Internet of Things Enabled Energy Aware Metaheuristic Clustering for Real Time Disaster Management. Comput. Syst. Sci. Eng., 45(2), 1561-1576.

28. Atsa’am, D. D., Gbaden, T., & Wario, R. (2023). A machine learning approach to formation of earthquake categories using hierarchies of magnitude and consequence to guide emergency management. Data Science and Management, 6(4), 208-213.

29. Saad, O. M., Huang, G., Chen, Y., Savvaidis, A., Fomel, S., Pham, N., & Chen, Y. (2021). Scalodeep: A highly generalized deep learning framework for real-time earthquake detection. Journal of Geophysical Research: Solid Earth, 126(4), e2020JB021473.

30. Bukhari, S. A. S., Shafi, I., Ahmad, J., Butt, H. T., Khurshaid, T., & Ashraf, I. (2024). Enhancing flood monitoring and prevention using machine learning and IoT integration. Natural Hazards, 1-28.

31. Damaševičius, R., Bacanin, N., & Misra, S. (2023). From sensors to safety: Internet of Emergency Services (IoES) for emergency response and disaster management. Journal of Sensor and Actuator Networks, 12(3), 41.

32. Devaraj, J., Ganesan, S., Elavarasan, R. M., & Subramaniam, U. (2021). A novel deep learning based model for tropical intensity estimation and post-disaster management of hurricanes. Applied Sciences, 11(9), 4129.