



Faculty of Cognitive Sciences and Human Development

**Performance Measurement on Deep Spiking Neural Network
(DSNN) Algorithm in Flood Prediction Environment**

Roselind Tei

**Master of Science
2023**

Performance Measurement on Deep Spiking Neural Network
(DSNN) Algorithm in Flood Prediction Environment

Roselind Tei

A thesis submitted

In fulfillment of the requirements for the degree of Master of Science

(Intelligent System)

Faculty of Cognitive Sciences and Human Development

UNIVERSITI MALAYSIA SARAWAK

2023

DECLARATION

I declare that the work in this thesis was carried out in accordance with the regulations of Universiti Malaysia Sarawak. Except where due acknowledgements have been made, the work is that of the author alone. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.



.....

Signature

Name: Roselind Tei

Matric No.: 18020147

Faculty of Cognitive Sciences and Human Development

Universiti Malaysia Sarawak

Date : 7 August 2023

ACKNOWLEDGEMENT

I want to extend my sincere gratitude to the remarkable individuals and institutions whose contributions played a pivotal role in the resounding success of my study work. Foremost, I am indebted to my esteemed supervisor, Dr. Abdulrazak Yahya Saleh, whose unwavering guidance, invaluable advice, and constant encouragement were instrumental throughout my master's program. His mentorship has shaped me into a better researcher and academic.

In addition, I express my profound thanks to the management of the Universiti Malaysia Sarawak and Zamalah Scholarship for their unwavering support and assistance. Their dedication to fostering an enriching academic environment has been a driving force behind my accomplishments.

Finally, I would like my gratitude knows no bounds when it comes to acknowledging the unwavering support and love of my parents and siblings. Their understanding, patience, and unconditional love have been my pillars of strength, providing the necessary emotional support throughout this journey. Thank you all.

ABSTRACT

There are several algorithms used to predict floods, including LSTM, BP, MLP, SARIMA, and SVM. While shallow neural networks are simple and efficient, they have limited memory and may not accurately capture long-term patterns or large-scale data. LSTM has gained attention among researchers in flood prediction for its ability to preserve historical data and solve complex time series problems. However, the study of this area is ongoing, with potential for further improvement. In current studies, researchers are exploring new directions by developing hybrid algorithms. The SNN, a third generation ANN, has been created to handle more complex data with a higher decision-making firing rate than ML and DL. In this study, a new hybrid DSSN algorithm will be utilised to predict floods in Kuala Baram, Miri, Sarawak. Rainfall data from 30 years (1989-2019) was collected from DID to evaluate the effectiveness of the DSSN algorithm compared to traditional and shallow neural networks algorithms. Performance was measured using ACC, RMSE, SPE, SEN, PPV, NPV, and ASP. A comprehensive analysis of the proposed DSSN algorithm was conducted. Four different training batch ratios were used to validate: 80:10:10, 70:15:15, 60:20:20, and 50:25:25. The results of the study showed that the DSSN algorithm outperformed the other algorithms with a higher ACC rate of 98.10%, an RMSE of 6.5%, a SEN of 93.50%, an SPE rate of 79.00%, and ASP of 89.60%. Overall, the DSSN algorithm with an 80:10:10 training sample ratio performed the best.

Keywords: Deep Spiking Neural Network (DSSN), Flood Prediction, Leaky Integrate and Fire (LIF), Long Short-Term Memory (LSTM), Spiking Neural Network (SNN).

Pengukuran Prestasi Algoritma Deep Spiking Neural Network (DSNN) dalam Ramalan Banjir

ABSTRAK

Terdapat beberapa algoritma yang digunakan untuk meramal banjir, termasuk LSTM, BP, MLP, SARIMA, dan SVM. Walaupun rangkaian neural adalah mudah dan cekap, ia mempunyai ingatan terhad dan mungkin tidak dapat mengesan corak jangka panjang atau data berskala besar secara tepat. LSTM telah menarik perhatian penyelidik dalam ramalan banjir kerana keupayaannya untuk mengekalkan data sejarah dan menyelesaikan masalah rangkaian masa yang kompleks. Walau bagaimanapun, kajian dalam bidang ini masih berterusan dengan potensi untuk peningkatan lebih lanjut. Dalam kajian semasa, para penyelidik sedang mengeksplorasi arah baru dengan membangunkan algoritma hibrid. SNN, generasi ketiga ANN, telah dicipta untuk mengendalikan data yang lebih kompleks dengan kadar keputusan yang lebih tinggi daripada ML dan DL. Dalam kajian ini, satu algoritma hibrid baru DSNN akan digunakan untuk meramal banjir di Kuala Baram, Miri, Sarawak. Data hujan selama 30 tahun (1989-2019) telah dikumpulkan dari DID untuk menilai keberkesanan algoritma DSNN berbanding dengan algoritma rangkaian neural tradisional. Prestasi diukur menggunakan ACC, RMSE, SPE, SEN, PPV, NPV, dan ASP. Analisis menyeluruh terhadap algoritma DSNN yang dicadangkan telah dijalankan. Empat nisbah latihan yang berbeza digunakan untuk pengesahan: 80:10:10, 70:15:15, 60:20:20, dan 50:25:25. Hasil kajian menunjukkan bahawa algoritma DSNN memberikan prestasi lebih baik berbanding algoritma lain dengan kadar ACC yang lebih tinggi, iaitu 98.10%, RMSE sebanyak 6.5%, SEN sebanyak 93.50%, kadar SPE sebanyak 79.00%, dan ASP sebanyak 89.60%. Secara keseluruhannya, algoritma DSNN dengan nisbah sampel latihan 80:10:10 memberikan prestasi terbaik.

Kata kunci: *Deep Spiking Neural Network (DSNN), Leaky Integrate and Fire (LIF), Long Short-Term Memory (LSTM), Ramalan banjir, Spiking Neural Network (SNN).*

TABLE OF CONTENTS

	Page
DECLARATION	i
ACKNOWLEDGEMENT	ii
ABSTRACT	iii
<i>ABSTRAK</i>	iv
TABLE OF CONTENTS	vi
LIST OF TABLES	x
LIST OF FIGURES	xi
LIST OF ABBREVIATIONS	xiii
CHAPTER 1: INTRODUCTION	1
1.1 Study Background	1
1.1.1 Flood in Baram, Miri	8
1.2 Problem Statement	11
1.3 Research Question	15
1.4 Research Objectives	15
1.5 Research Scope	15
1.6 Thesis Outlines	16

CHAPTER 2: LITERATURE REVIEW	18
2.1 Overview	18
2.2 Deep Learning (DL)	19
2.2.1 Long Short-Term Memory (LSTM)	22
2.2.1.1 Sigmoid Layer	24
2.2.1.2 Forget Gate	25
2.2.1.3 Input Gate and Tanh Layer	26
2.2.1.4 Update Cell State	27
2.2.1.5 Output Gate	28
2.3 Spiking Neural Network (SNN)	29
2.3.1 Leaky Integrate and Fire (LIF)	30
2.4 Related Work and Discussion	32
2.4.1 Comprehensive Literature Review of DL and ML in Flood Prediction	
Related Works	33
2.4.2 Comprehensive Literature Review of DL and ML in Flood Application	
Related Works	49
2.4.3 Comprehensive Literature Review of Hybrid DL and ML in Flood	
Prediction Related Works	59
2.5 Summary	69

CHAPTER 3: RESEARCH METHODOLOGY	70
3.1 Overview	70
3.2 General Research Framework	70
3.2.1 Phase 1: Data Preparation	71
3.2.1.1 Data Collection	72
3.2.1.2 Data Pre-Processing	73
3.2.2 Phase 2: Experimental Design	76
3.2.2.1 Hybridisation of SNN with LSTM (DSNN)	76
3.2.2.2 Experimental Settings	81
3.2.3 Phase 3: Evaluation and Comparison	83
3.2.3.1 Performance Measures	83
3.3 Summary	86
CHAPTER 4: RESULTS, ANALYSES AND FINDINGS	87
4.1 Overview	87
4.2 Proposed DSNN Algorithm Findings	87
4.2.1 Result Ratio (80:10:10)	88
4.2.2 Result Ratio (70:15:15)	90
4.2.3 Result Ratio (60:20:20)	93
4.2.4 Result Ratio (50:25:25)	96
4.3 Comparative Study and Analysis	100

4.3.1	Accuracy (ACC%)	100
4.3.2	RMSE Values (%)	102
4.3.3	Sensitivity (SEN%)	104
4.3.4	Specificity (SPE%)	106
4.3.5	Positive Predictive Value (PPV%)	108
4.3.6	Negative Predictive Value (NPV%)	110
4.3.7	Average Site performance (ASP%)	112
4.4	Performance Analysis	114
4.5	Summary	116
CHAPTER 5: CONCLUSION AND RECOMMENDATIONS		117
5.1	Conclusion	117
5.2	Summary of the Research	117
5.3	Research Contributions	117
5.4	Research Limitations	118
5.5	Future Works	119
5.6	Summary	119
REFERENCES		120
APPENDICES		148

LIST OF TABLES

	Page
Table 1.1 Details of Rainfall Classification of Warning Level	8
Table 2.1 Summary of DL and ML in Flood Prediction Related Works	33
Table 2.2 Summary of DL and ML in Flood Application Related Works	49
Table 2.3 Summary of Hybrid DL and ML in Flood Prediction Related Works	59
Table 3.1 Rainfall of Basin Stations Kuala Baram, Miri, Sarawak	72
Table 3.2 Overview Ratio used in this Research	81
Table 3.3 Batch Size, Epoch and Validation Steps Hyper Parameters of DSNN Algorithm	82
Table 3.4 Activation and Optimisation Hyper Parameters of DSNN Algorithm	82
Table 4.1 Result Ratio (80:10:10)	88
Table 4.2 Result Ratio (70:15:15)	90
Table 4.3 Result Ratio (60:20:20)	93
Table 4.4 Result Ratio (50:25:25)	96
Table 4.5 Accuracy (ACC%)	100
Table 4.6 RMSE Values (%)	102
Table 4.7 Sensitivity (SEN%)	104
Table 4.8 Specificity (SPE%)	106
Table 4.9 Positive Predictive Value (PPV%)	108
Table 4.10 Negative Predictive Value (NPV%)	110
Table 4.11 Average Site performance (ASP%)	112
Table 4.12 Performance Analysis	114

LIST OF FIGURES

	Page
Figure 1.1 Baram River Flow around the Miri, Sarawak	4
Figure 1.2 Topographic Map of Kuala Baram, Miri in 1960	5
Figure 1.3 Miri City in 1960	5
Figure 1.4 Batang Baram Anchorage Area	6
Figure 1.5 Baram River Baram Basin Location in Baram, Miri	7
Figure 1.6 Flash Floods Strike Kuching City, Kota Samarahan	9
Figure 1.7 Flash Floods Strike Residents of Batu Kawa, Kuching	9
Figure 1.8 The Road is Submerged by the Flash Flood	10
Figure 1.9 Property Damage due to sudden Floods	11
Figure 2.1 The Learning Structure of Deep Learning (DL)	21
Figure 2.2 Overview Schematic of the Simple Recurrent Network	23
Figure 2.3 Long-Term Short-Term Architecture Model (LSTM)	24
Figure 2.4 Horizontal Line Running Through the Top of LSTM Cell State	24
Figure 2.5 Sigmoid Layer in LSTM	25
Figure 2.6 LSTM “Forget Gate”	26
Figure 2.7 LSTM “Input Gate and Tanh Layer”	27
Figure 2.8 LSTM “Update Cell State”	28
Figure 2.9 LSTM “Output”	29
Figure 2.10 Diagram of the LIF	31
Figure 3.1 Framework of the Study	71
Figure 3.2 Simple Schematic Representation of DSNN Proposed Method	78
Figure 3.3 Pseudo-code of Hybrid DSNN	79
Figure 4.1 Performance Measurement with Ratio (80:10:10)	88

Figure 4.2	Performance Measurement with Ratio (70:15:15)	90
Figure 4.3	Performance Measurement with Ratio (60:20:20)	94
Figure 4.4	Performance Measurement with Ratio (50:25:25)	97
Figure 4.5	Accuracy Results (ACC%)	101
Figure 4.6	RMSE Values Results (%)	103
Figure 4.7	SEN Results (%)	105
Figure 4.8	SPE Results (%)	106
Figure 4.9	PPV Results (%)	108
Figure 4.10	NPV Results (%)	110
Figure 4.11	ASP Results (%)	112
Figure 4.12	Total Performance Results	114

LIST OF ABBREVIATIONS

ACC	Accuracy
ACF	Auto-Correlation Function
AER	Address Event Representation
AEs	Autoencoders
AHP	Analytic Hierarchy Process
ANN	Artificial Neural Network
ARIMA	Autoregressive Integrated Moving Average
ARMA	Autoregressive Moving Average
ASP	Average Site Performance
ASTER	Advanced Spaceborne Thermal Emission and Reflection Radiometer
AUC	Area Under the Curve
BILSTM	Bidirectional Long Short-Term memory
BN	Bayesian Networks
BP	Backpropagation
CC	Coefficient Correlation
CGS	Centre for Graduate Studies
CNN	Convolutional Neural Network
ConvNets	Convolutional Neural Networks
CRBM	Conditional Restricted Machine
DBN	Deep Belief Network
DEM	Digital Elavation Model

DID	Department of Irrigation and Drainage
DL	Deep Learning
DRNN	Deep Recurrent Neural Networks
DSSN	Deep Spiking Neural Network
DSLML	Deep Liquid State Machine
DVS	Dynamic Vision Sensor
EBF	Evidential Belief Function
ELM	Extreme Learning Machine
FN	False Negative
FP	False Positive
FR	Frequency Ratio
FRCBM	Factored Conditional Restricted Boltzmann Machine
GEE	Google Earth Engine
GFI	Geomorphic Flooding Index
GIS	Geographic Information System
GPS	Global Positioning System
GRU	Gated Recurrent Unit
GSM	Global System for Mobile Communications
IoT	Internet of Things
KNN	K-Nearest Neighbours
Landsat TM	Landsat Thematic Mapper
LeNet-5 CNN	LeNet – 5 Convolutional Neural Network
LIF	Leaky Integrate and Fire
LME	Legate McCabe Efficiency Index
LoRaWan	Long Range Wide Area Network

LSTM	Long Short-Term Memory
LULC	Land Use and Land Cover
MAE	Mean Absolute Error.
MAPE	Mean Percentage Error
MDEL	Multiscale Deep Feature Learning
ML	Machine Learning
MLP	Multilayer Perceptron
MSE	Mean Squared Error
NARX	Nonlinear AutoRegressive with Exogenous inputs
NDWI	Normalised Difference Water Index
NM	Numerical Model
NPV	Negative Predictive Value
NSE	Nash-Sutcliffe Efficiency
OLI	Operational Land Image
OWA	Ordered Weight Average
PACF	Partial Auto-Correlation Function
PC	Personal Computer
PDBM	Predictive Boltzmann Machine
PPV	Positive Predictive Value
PSO	Particle Swam Optimization
r	Coefficient Correlation
R ²	R-Squared
RAM	Random Access Memory
RBFNN	Radial Basis Function Neural Network
RBM	Restrictive Boltzmann Machine

ResNet	Residual Neural Network
RGB	Red, Green, Blue
RMSE	Root Mean Square Error
RNN	Recurrent Neural Networks
ROC	Receiver Operating Characteristic
RWL	Reservoir Water Level
SAE	Stacked Autoencoders
SAR	Synthetic Aperture Radar
SARIMA	Seasonal Autoregressive Integrated Moving Average
SDAE	Stacked Denoising Auto Encoder
SDWI	Shadow Difference Water Index
SEN	Sensitivity
SMS	Short Message Service
SNN	Spiking Neural Network
SPE	Specificity
SRM	Spike Respond Model
SVM	Support Vector Machine
TN	True Negative
TNR	True Negative Rate
TP	True Positive
TRP	True Positive Rate
UNIMAS	Universiti Malaysia Sarawak
VE	Validation Epoch
VGG Net	Visual Geometry Group Network
WOA	Whale Optimisation Algorithm

XAJ

Xinjiang Model

CHAPTER 1

INTRODUCTION

1.1 Study Background

In 2021, Malaysia's massive floods resulted in an estimated total loss of RM6.1 billion, with Selangor being the worst hit state (Abdullah, 2018; Bedi, 2022; Mohammed et al., 2018; Safiah Yusmah et al., 2020). According to an article in the New Straits Times by Dr. Mohd Uzidin (Bernama, 2022), the annual floods have adverse impacts on various aspects of life, including loss of lives, destruction of public assets, houses, transportation, infrastructure, manufacturing, and agriculture industries. These floods cause financial losses not only to business operators but also to citizens in general. The total loss due to floods adds up to more than RM2 billion, with houses (RM1.6 billion), automobiles (RM1.0 billion), the manufacturing sector (RM0.9 billion), commercial real estate (RM500 million), and agriculture (RM90.6 million) being the most affected.

Flood is a natural disaster that hinders a country's development (Elias et al., 2013; Kourgialas & Karatzas, 2011). Flooding typically happens when swollen water from rivers or lakes overflows the levees, but it can also occur when rainwater collects on saturated ground with insufficient areas for infiltration. Some floods develop slowly, whereas flash floods can happen very fast. Floods are becoming more frequent due to climate change, and transformation of the earth's physical infrastructure as a consequence of technical and economic improvements (Elias et al., 2013). Serious attention must be given to flood control measures and solutions developed to mitigate the negative impact of this disaster. Past research has run several statistical tests and analyses in Malaysia, including Sabah and Sarawak; the results show that since the 1920s, floods have been the most common natural

calamity, hitting many locations, particularly the low-lying areas (Pirah & Roslee, 2021; Tew et al., 2022). The expansion of the areas causes almost 9% of the entire disaster, while floods bring direct damages to nearly 22% of the total population (Yusoff et al., 2018). Furthermore, Malaysia's climate is experiencing an increase in the frequency of rainfall, with an average of roughly 2,500 mm per year in Peninsula Malaysia, 3,000 mm per year in Sabah, and 3,500 mm per year in Sarawak (Aliagha et al., 2015; Othman et al., 2014; Salleh et al., 2013; Tan et al., 2015).

During the monsoon seasons, most Malaysian states receive considerable rainfall every year, resulting in flooding in numerous locations, particularly those in and around central business districts. The district of Baram River in Sarawak is vulnerable to flooding due to its proximity to the river mouth fronting the South China Sea. The Baram River is the second longest river in Sarawak with its source located hundreds of kilometres inland in the mountains near the Sarawak-Indonesia border (Ling et al., 2017). The river begins in Long Lamai and ends at Baram with water flowing into the South China Sea (KessLer & Jong, 2015). Few studies on flood event prediction have used physical modelling techniques such as regression (Fenglin et al., 2023; Panahi et al., 2021; Rezaeianzadeh et al., 2018; Tsakiri et al., 2018; Viteri López & Morales Rodriguez, 2020) and hydrodynamic modelling (Boota et al., 2023; Fitzpatrick et al., 2023; Huang et al., 2023; Karim et al., 2023; Timbadiya & Krishnamraju, 2023).

High-performance models can assist in predicting impending flooding events and mitigate the damaging effects. One of the most effective techniques is to use computer modelling. Computer modelling has been frequently utilised in research to tackle physical difficulties. Researchers can easily and quickly evaluate changes in real-life situations and

their impacts over time. With the advent of high-speed computers and sophisticated software packages, many digitalised models have been developed to forecast outcomes of floodings (Fotovatikhah et al., 2018; Jajarmizadeh et al., 2014; Teng et al., 2017).

Deep Artificial Neural Network (ANN), also known as Deep Learning (DL), is the most well-known computerised modelling that solves problems using algorithmic techniques (Schmidhuber, 2015). Shallow and deep neural networks, with different credit assignment paths, link actions and outcomes utilising pattern recognition, generalisation, perception, and machine learning. DL consists of deep supervised learning, which encompasses recapitulating the history of Back-propagation, unsupervised learning, reinforcement learning and evolutionary computation, and indirect search for short programs encoding deep and large networks (Hinton, 2007; LeCun et al., 2015; Liu et al., 2017). Spiking Neural Network (SNN), the third generation of Artificial Neural Network (ANN), plays a vital role in biological information processing (Maass, 1997). The SNN model provides an in-depth description of biological neuronal behaviour. Researchers have used additional data to analyse the average firing rate for neuron computations. The structure of SNNs is optimised for the efficient processing of firing times; the transmission of information between neurons occurs via action potentials or spikes.

In contrast, ANN models average individual spikes over time, dividing all interactions by the average neuron firing rate (Ikeda & Manton, 2009). They are also more computationally robust than ANN, which uses a mean firing rate (Ignatov et al., 2015). Therefore, due to the robustness of DL and SNN models, this study fits and integrates the DL model's spike trains and hybrid training into Deep Spiking Neural Network (DSNN).

The study area is the Baram River, in Miri Division of Sarawak, and north-western Borneo. The Baram River flows around the Miri District, as shown in Figure 1.1 shows Baram river flow around the Miri, Sarawak. Figure 1.2 illustrate topographic map of Kuala Baram, Miri in 1960. Meanwhile, Figure 1.3 illustrates the city of Miri city in 1960. The Baram River basin, covering an area of approximately 10,000 square miles, has been a part of Sarawak since it was surrendered to the White Rajah of Sarawak by the then Sultan of Brunei in 1882. (30,000 km²). Figure 1.4 shows a closer view of the Batang Baram anchorage area, and Figure 1.5 shows the location of the Baram Miri Basin.



Figure 1.1: Baram River Flow around the Miri, Sarawak (Dandot, 2006)

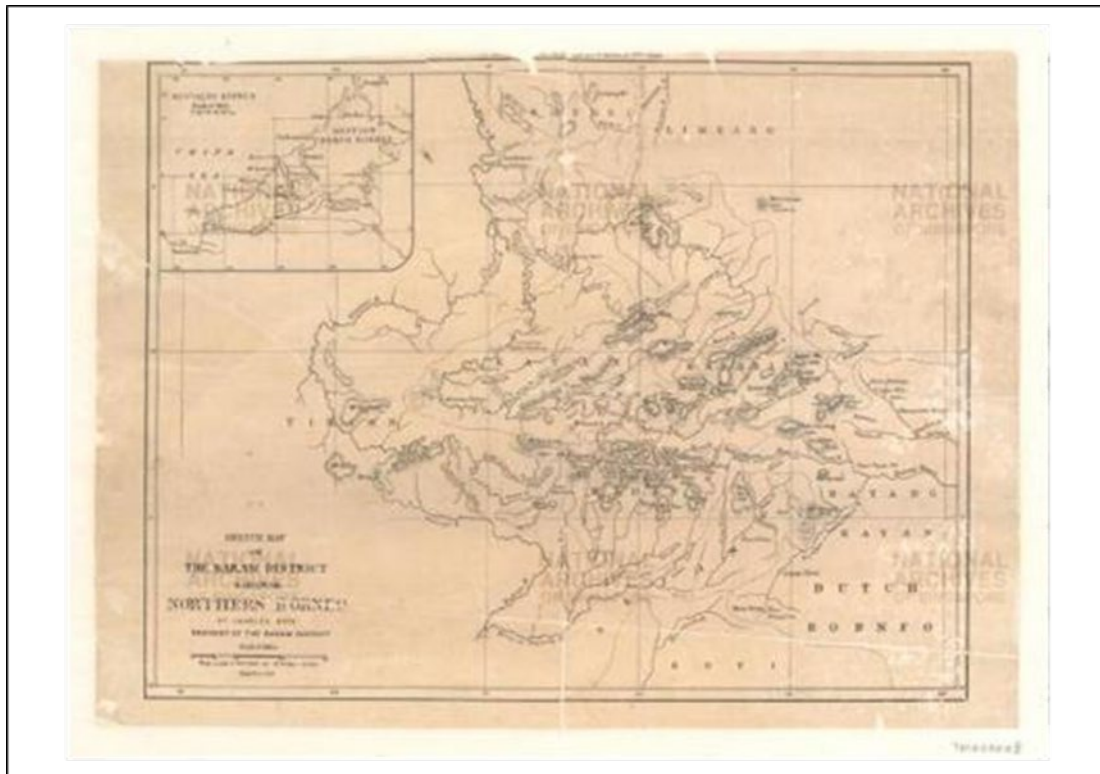


Figure 1.2: Topographic Map of Kuala Baram, Miri in 1960 (Dandot, 2006)

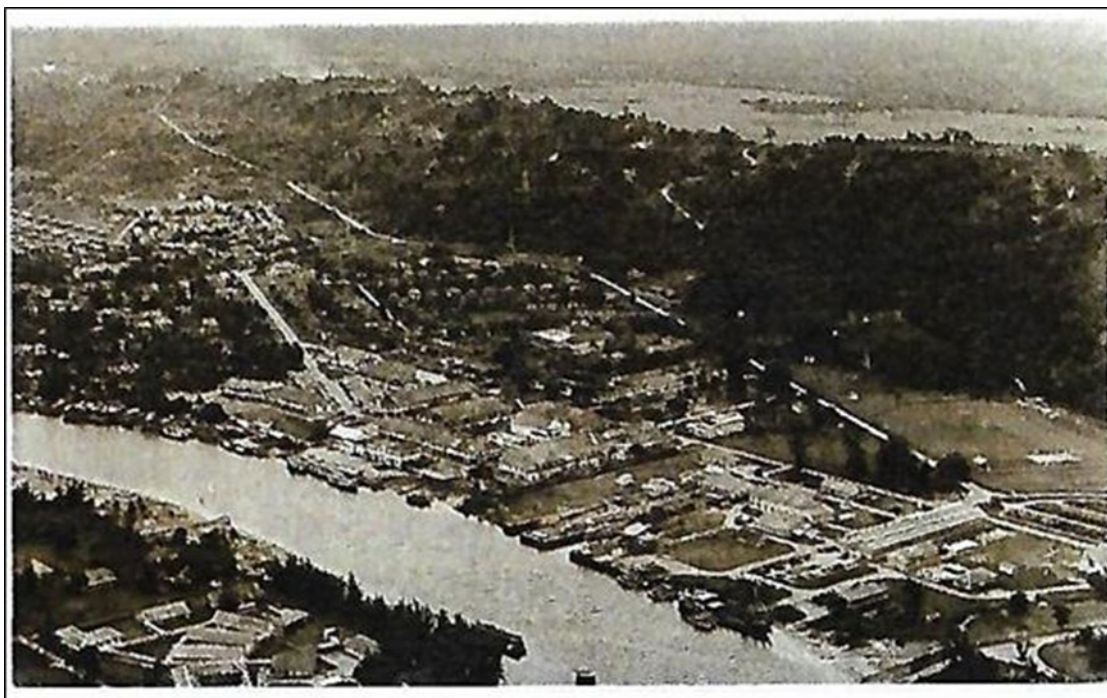


Figure 1.3: Miri City in 1960 (Dandot, 2006)



Figure 1.4: Batang Baram Anchorage Area (Sarawak River Board, 2018)

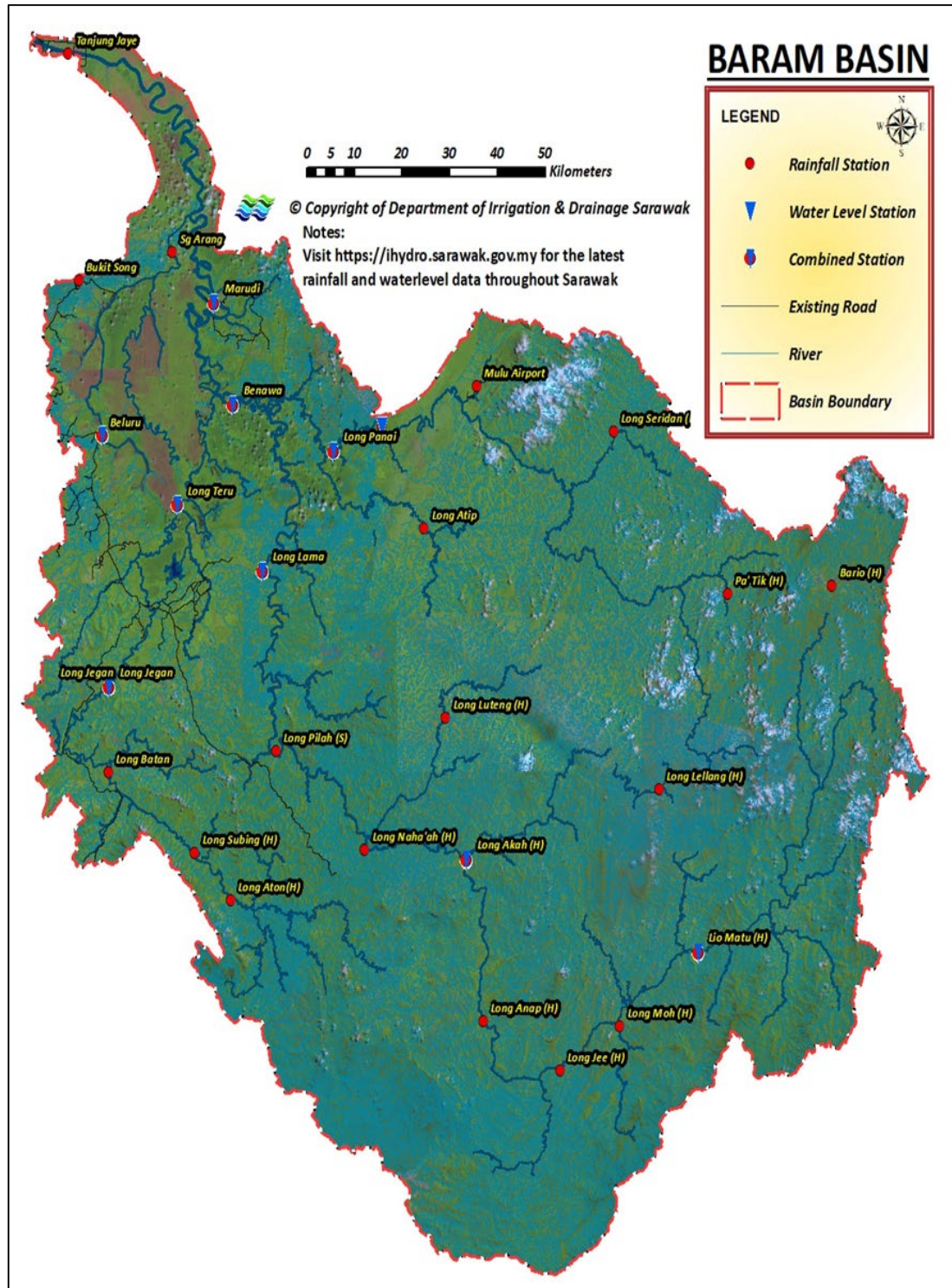


Figure 1.5: Baram River Baram Basin Location in Baram, Miri (Department of Irrigation and Drainage Sarawak, 2018)

1.1.1 Flood in Baram, Miri

The Department of Irrigation and Drainage (DID) classifies rainfall intensity (in one hour) into four categories: average, alert, warning, and danger. Table 1.1 shows more details, and that the river's alert level has significantly increased above the normal. The warning category shows the river level is increasing to near flooding stage and be prepared for any evacuation action. The danger category indicates imminent flooding, and immediate evacuation needs to be initiated (The Official Web of Public Infobanjir, 2018).

Table 1.1: Details of Rainfall Categorization of warning level

Warning Level	Categorization of Rainfall Intensity in one hour/mm	Color
Light	1 - 10 (mm)	
Moderate	11 - 30 (mm)	
Heavy	30 – 60 (mm)	
Very Heavy	> 60 (mm)	

A recent rainstorm in March 2023 caused severe flooding in Kuching, Miri, rural Sarawak, and even West Malaysia. Homes were inundated by floods no less than five times in less than three months. Roads, hospitals, and fire stations were all flooded, making it difficult for the rescue teams to relocate flood victims to safer and temporary places (Lai, 2023; Ringgit, 2022). Figure 1.6 shows the flash floods striking Kuching City, Kota Samarahan. Meanwhile, another flash flood strikes the residents of Batu Kawa, Kuching, as shown in Figure 1.7.



Figure 1.6: Flash Floods Strike Kuching City, Kota Samarahan (Samual, 2023)



Figure 1.7: Flash Floods Strike Residents of Batu Kawa, Kuching (Roji Kawi, 2023)

The Borneo Post reported on Thursday, 12th July 2018: flash floods again hit Kuala Baram early in the year (Abdullah, 2018). A thunderstorm pounded the city on Tuesday night, resulting in a flash flood; parents had problems in sending their children to school, as shown in Figure 1.8.



Figure 1.8: The Road is Submerged by the Flash Flood (Abdullah, 2018)

The Star Online News reported on Tuesday, February 28, 2017: due to extensive flooding, over 10,000 indigenous people and school children living in at least 15 longhouses in the remote Baram region of northern Sarawak were cut off from the outside world. Torrential rain had overwhelmed the towns over the last 24 hours, with water levels reaching more than a meter high (Stephen, 2017). Meanwhile, on Wednesday, December 20, 2017, The Star Online News reported: flash floods hit northern Sarawak, Kuala Baram, and Miri as a result of nonstop rain that began on late Tuesday night, December 19, 2017; it persisted into the early hours of Wednesday, causing seawater to overflow and flood many coastal settlements in the northern Sarawak. According to the Fire Department, hundreds of families

were relocated temporarily from their homes to higher grounds due to sudden floods, as shown in Figure 1.9 (Stephen, 2018).



Figure 1.9: Property Damage due to sudden Floods (Stephen, 2018)

1.2 Problem Statement

Flood prediction is critical in mitigating the impacts of one of nature's most destructive forces (Ighile et al., 2022; Kumar et al., 2023a; Tillihal & Shukla, 2023). Accurate predictions of approaching floods enable the relevant authorities to take necessary actions: proactive emergency response, informed decision-making, and the implementation of preventive measures to safeguard lives, protect infrastructure, and minimise financial losses. To achieve this, it is essential to develop and improve high-performance flood prediction methods (Kim et al., 2023).

Several methods of flood prediction have been developed in recent decades, such as physical models, hydrodynamic models, and regression models (Anderson, 2002; Chang et

al., 2018; Gao et al., 2023; Mosavi et al., 2018). However, implementing these methods are costly and time-consuming due to the need for extensive data collection (Chu et al., 2020; Kumar et al., 2023b; Luo et al., 2022; Wu et al., 2020; Zhou et al., 2023).

Due to the vast benefits and potential, Machine Learning (ML) has been increasingly popular among researchers. Shallow neural networks and traditional algorithms are widely used in flood predictions, which include Backpropagation (BP), Multilayer Perceptron (MLP), Seasonal Autoregressive Integrated Moving Averages (SARIMA), and Support Vector Machines (SVM). Among various ANNs, backpropagation (BP) is the most commonly used prediction algorithm for forecasting flood time series (Dai & Cai, 2021; Paul & Das, 2014; Valipour et al., 2013). Meanwhile, the MLP algorithm has been widely used in flood prediction due to its simplicity, efficiency, and strong generalisation ability compared with other ANN classes (Riad et al., 2004; Rumelhart et al., 1986; Senthil Kumar et al., 2005). In addition, the SARIMA algorithm has the advantage of requiring few parameters to describe time series that show non-stationarity, both within and through seasons, and is widely used in water level prediction (Azad et al., 2022; Mirzavand & Ghaznavid, 2015). SVM is greatly popular in flood modelling; and is known for its linear and nonlinear classification and the efficient mapping of inputs into feature spaces. Thus, SVM has been applied in numerous flood prediction cases with promising results, excellent generalisation ability, and better performance (Dehghani et al., 2014; Dibike et al., 2001; Vapnik & Mukherjee, 1999; Yu et al., 2017). The applications of the above methods have yielded significant improvements in predicting water levels. However, shallow neural networks lack memory capacity and cannot efficiently capture long-term patterns and large-scale data (Asadi & Regan, 2020; Hosein & Hosein, 2017). In recent years, deep neural networks (also known as deep learning) have improved the performance of shallow neural

networks, including recurrent neural networks (RNNs) (Hinton et al., 2006; Oh et al., 2019). The extended short-term memory network (LSTM) (Hochreiter & Schmidhuber, 1997) uses memory blocks to store information for arbitrary periods, enabling long-term temporal correlations to be captured without optimisation hurdles, ultimately leading to better prediction accuracy (Yang et al., 2020).

The continuous development of prediction methods (Baek et al., 2020; Bai et al., 2021; Li et al., 2022; Nie et al., 2017), especially the LSTM (Long Short-term Memory) neural network, has enabled the forecasting procedures to store historical information by cyclic feedback while solving complex problems related to time series. Recently, there has been growing attention towards utilising LSTM for flood prediction. Research conducted by Liang et al. (2018) demonstrates that LSTM algorithms surpass SVM algorithms in forecasting water levels. Additionally, (Guo & Lai, 2020) found that LSTM can achieve greater accuracy in predicting water levels when more data are available.

A recent research trend points to a new development: machine learning involves the crafting of hybrid algorithms that synergistically combine two or more algorithms to perform specific tasks. Numerous studies have employed hybrid algorithms for flood prediction. However, this area of research continues to be actively explored and holds significant potential for further experimentation (Bayat & Tavakkoli, 2022; Bellos & Tsakiris, 2016; Li & Willems, 2020; Maier et al., 2010; Mosavi et al., 2018; See & Openshaw, 2000).

The SNN algorithm is integrated into LSTM and forms the hybridisation of DSNN algorithm to improve the performance of LSTM in flood prediction, by determining the optimal input hyperparameters and batch ratio (Ke & Vikalo, 2021; Keong et al., 2016; Singh & Kasana, 2022; Skrobek et al., 2020), batch size and epoch (Ahmed et al., 2021; Ding et

al., 2020; Hashemi et al., 2022; Poornima & Pushpalatha, 2019; Song et al., 2019; Xiang et al., 2020; Xu et al., 2021).

DSNN algorithms can detect long-term dependencies and temporal dynamics in sequential data, and are particularly beneficial in flood prediction tasks, where historical patterns and trends play a crucial role. The LSTM component of the hybrid algorithm allows it to retain and process information over longer sequences, enabling a more accurate representation of temporal dynamics, and capturing complex patterns of flood data (Hochreiter & Schmidhuber, 1997). LSTM is known for capturing complex linear and non-linear relationships in data (Bai et al., 2021; LeCun et al., 2015; Schmidhuber, 2015). By combining DL with SNN, the hybrid model can leverage the strengths of both approaches. LSTM captures long-term dependencies and non-linear relationships (Chitra & Rajasekaran, 2022; Ehteram & Ghanbari, 2023; Ian et al., 2023; Thamilselvan et al., 2022), while SNN improves the representation of spatiotemporal patterns and also captures linear and non-linear interactions in the data which create a hierarchical structure (layers) and subsequently construct a network (Geminiani et al., 2019; Yu et al., 2022). This combination allows for more comprehensive modelling of the underlying dynamics of flood events.

Therefore, the DSNN algorithm will be utilised in this study to achieve higher accuracy compared with shallow neural networks and traditional algorithms. Unlike many previous studies that mainly focused on accuracy (ACC) and Root Mean Square Error (RMSE) as the primary measure of algorithm performance (Barzegar et al., 2021; Xie et al., 2021; Wang et al., 2023), this study aims to comprehensively assess the performance of the DSNN algorithm by including additional metrics such as Accuracy (ACC), Root Mean Square Error (RMSE), and confusion matrix such as Sensitivity (SEN), Specificity (SPE),

Positive Predictive Value (PPV), Negative Predictive Value (NPV), and Average Site Performance (ASP) (Ballabio et al., 2018; Fawcett, 2006; Krstinić et al., 2020; Labatut & Cherifi, 2011).

1.3 Research Question

The following are the research questions addressed in this study:

- i. What methods can be utilised to pre-process the flood datasets?
- ii. What methods can be used to propose integrating Spiking Neural Networks (SNN) with Deep Learning (DL) algorithms for predicting floods?
- iii. What methods can be utilised to evaluate the performance effectiveness of the Deep Spiking Neural Network (DSNN) algorithm compared with that of the shallow neural networks and traditional algorithms?

1.4 Research Objectives

The objectives of this research are identified as below:

- i. To pre-process the rainfall datasets for the Deep Spiking Neural Network (DSNN) algorithm.
- ii. To propose a new DSNN algorithm for flood prediction.
- iii. To analyse the effectiveness of the DSNN algorithm compared with that of the shallow neural networks and traditional algorithms.

1.5 Research Scope

The scope of this research is limited to the following to achieve the desired objectives:

- i. The purpose of this study is to formulate a procedure for flood prediction.

- ii. The area of study is Kuala Baram, located in Miri Division, Sarawak.
- iii. Thirty years of historical rainfall datasets (1989–2019) were collected from DID.
- iv. The Deep Spiking Neural Network (DSNN) algorithms are applied for flood prediction and the results are compared with those of the shallow neural networks and traditional algorithms.
- v. Performance is tested based on Accuracy (ACC), Root Mean Square Error (RMSE), Specificity (SPE), Sensitivity (SEN), Negative Predictive Value (NPV), Positive predictive Value (PPV), and Average Site Performance (ASP).
- vi. The Python programming language is utilised in this research.

1.6 Thesis Outline

Chapter 1 provides the background information of the Baram River area, including nature of problems, problem statement, objectives, scope, and research significance.

Chapter 2 reviews previous works done in flood prediction, and related algorithms used, including Long Short-Term Memory (LSTM) and the Leaky Integrate and Fire (LIF) Model of the Spiking Neural Network (SNN).

Chapter 3 explains the methodology of this research, including the proposed DSNN algorithm, hyperparameters, considerations for model training, and computation of performance output, such as Accuracy (ACC), Root Mean Square Error (RMSE), and confusion matrices.

Chapter 4 analyses the performance of the proposed DSNN algorithm in classifying flood data based on measures such as ACC, RMSE, SPE, SEN, NPV, PPV, and ASP.

Chapter 5 concludes the findings of this study and future works.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview

Integrating Deep Learning (DL) with Spiking Neural Network (SNN), the hybrid algorithm effectively capitalises on the strengths of both methodologies. LSTM excels at capturing long-term dependencies and non-linear relationships. SNNs have attracted a significant researcher interest in recent years due to their prospective properties, such as the ability to process temporal information (Petro et al., 2019), low power consumption (Roy et al., 2019), and biological interpretability (Gerstner et al., 2014). Currently, SNNs can process event stream data with low latency and low power (Gallego et al., 2020; Pei et al., 2019). However, a performance gap exists between SNNs and traditional Artificial Neural Networks (ANNs). Recent SNN training techniques based on gradients and back-propagation have greatly improved the performance of SNNs (Fang et al., 2021; Wu et al., 2021). Therefore, this study integrates DL which better performance in data complexity compared to shallow neural networks or traditional algorithms to integrate into SNN to achieve higher accuracy in flood prediction. This unique fusion enables more comprehensive modelling of the underlying dynamics of flood events.

The performance of the DSNN algorithm can be assessed using multiple metrics such as Accuracy (ACC), Root Mean Square Error (RMSE), and confusion matrix measures such as Sensitivity (SEN), Specificity (SPE), Positive Predictive Value (PPV), Negative Predictive Value (NPV), and Average Site Performance (ASP). Unlike previous studies that primarily focused on Accuracy (ACC) and Root Mean Square Error (RMSE), this study aims to comprehensively evaluate the performance of the DSNN algorithm by including

additional metrics. Chapter 2 thoroughly examines past study done on the prediction field utilising ML, DL approaches. This review offers a comprehensive analysis of previous literature on related work.

2.2 Deep Learning (DL)

Different researchers have widely applied DL to conduct studies related to flood prediction (Cho et al., 2014; Dubey, 2015; Kuok et al., 2011; Kuok et al., 2010; Liu et al., 2017; Saha et al., 2017; Saha & Mitra, 2016; Salman et al., 2015). DL has been developed into different architecture models: supervised, unsupervised, and reinforcement learning, as shown in Figure 2.1. Several DLs can operate with both supervised and unsupervised learning. DL has multiple hidden layers and a different architecture (Castrounis, 2016). DL can learn and implement a large amount of data in the industrial area. It uses supervised learning to guide the computer as it processes thousands or millions of pieces of data. In this study, flood data are categorized as a large quantity of data; therefore, deep learning is the suitable neural network to process the large volume of flood data and provide accurate algorithms that can predict the occurrence of floods.

Geoffrey Hinton first introduced DL in his paper "Learning Multiple Layers of Representation" in the middle of the 2000s (Hinton, 2007). This method is based on the human brain's ability to solve complex problems through abstraction. DL can be trained to solve simple and complex problems by traversing through multiple layers of information. Unlike ANN, DL can handle large amounts of data that require extensive processing. It consists of input, hidden, and output layers.

Through supervised learning, DL can process vast amounts of data in the industrial sector. This makes DL the ideal neural network to handle the large volume of flood data and

produce accurate algorithms for flood prediction. Therefore, DL is the appropriate approach to accurately predict floods by generating algorithms based on the large volume of data.

This section introduces widely used DL models known as 'classical' and provides a quick overview. A significant advancement in deep learning training, as opposed to shallow networks with three or fewer layers, was the layer-wise approach suggested for training stacked Restrictive Boltzmann Machines (RBM) by Hinton in 2007. This strategy has also effectively trained other types of deep networks, such as stacked Autoencoders (Bengio et al., 2006). RBMs and AEs are suitable for dimensionality reduction and feature extraction when trained with unsupervised learning on unlabelled data. The Google Brain team accomplished a DL milestone in 2012 by utilising an unsupervised DL model. Through analysing 10 million photos extracted from various frames of YouTube videos, the neural network developed the ability to differentiate cats (Le, 2013).

Hubel and Wiesel (1962) seminal finding that orientation selectivity (simple cells) and pooling mechanisms (complex cells) represent the core activities in the primary visual cortex of cats inspired ConvNets. These simple cells respond to their preferred orientation of visual stimuli within their receptive fields, which are small sub-regions of the visual field, at a high frequency. In the meantime, a complex cell correlates to the presence of a pattern within a broader receptive field but loses the pattern's precise position.

NeoCognitron (Fukushima & Miyake, 1982) was the first ANN model to mimic the functions of V1 simple and complex neurons. The ability of single cells to recognize features was further improved by sharing weights among receptive fields in LeNet-5, the most common ConvNet used today. In a ConvNet, convolution involves weight sharing, significantly reducing the number of weight parameters within a network. Deep ConvNet's

most significant milestones dominated the best performances in the yearly ImageNet Challenge (Russakovsky et al., 2015), VGG Net (Simonyan & Zisserman, 2014), AlexNet (Krizhevsky et al., 2012), GoogLeNet (Szegedy et al., 2015), and ResNet (He et al., 2015). Despite their strong capabilities, these feed-forward deep networks require assistance with sequence processing because the number of neurons limits the size of the input and output vectors. As a result, Recurrent Neural Networks (RNNs) with feedback connections are perfect for dealing with sequential information because their current output is always dependent on the previous memory. As training techniques have matured, such as using Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997), RNNs have shown great success in many natural language processing tasks: language modelling, machine translation, speech recognition (Graves & Jaitly, 2014), and image caption generation (Karpathy & Fei-Fei, 2015).

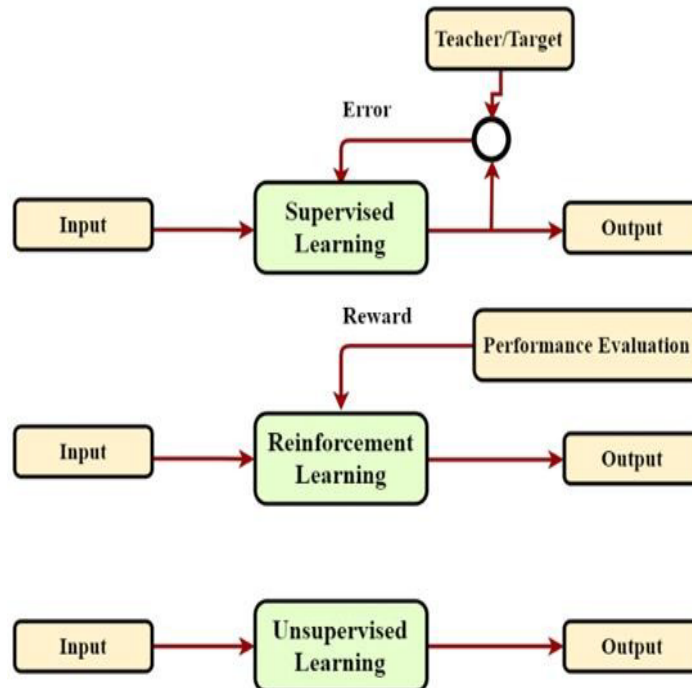


Figure 2.1: The Learning Structure of Deep Learning (DL) by Chen (2017)

2.2.1 Long Short-Term Memory (LSTM)

Recurrent networks can use their feedback connections to store representations of recent input events as activations from short-term memory rather than slowly embodied long-term memory. RNN has the potential to be significant for a wide range of applications, including speech processing, non-Markovian control, music composition, and prediction (Hochreiter & Schmidhuber, 1997).

The LSTM architectural model is a memory cell that can keep its state. The non-linear units of LSTM control the flow of information into and out of the cell. The most recent research has integrated numerous advancements from the original formulation of the LSTM architectural model. LSTM is currently used for a wide range of learning tasks that differ in scale and character from the issues on which these advancements were first tested (Greff et al., 2016; Kong et al., 2018; Selvin et al., 2017; Zhang et al., 2017).

Figure 2.2 depicts a schematic of the LSTM architecture model block. It has three gates: input, forget, output, block input, a Constant Error Carousel cell, an output activation function, and peephole connections. The block's output is recurrently coupled to the block's input, as are all gates (Greff et al., 2016; Selvin et al., 2017). In addition, LSTM techniques have been used in numerous research for flood prediction purposes. Therefore, due to the advantages shown in numerous research, LSTM will be adopted in this study (Duan et al., 2016; Gers et al., 2002; Jin et al., 2018; Marino et al., 2016; Roondiwala et al., 2017; Selvin et al., 2017; Song et al., 2018; Sugiartawan et al., 2017; Yang, 2017; Zaytar & Amrani., 2016; Zazo et al., 2016; Zhang et al., 2017; Zhuang & Ding 2016).

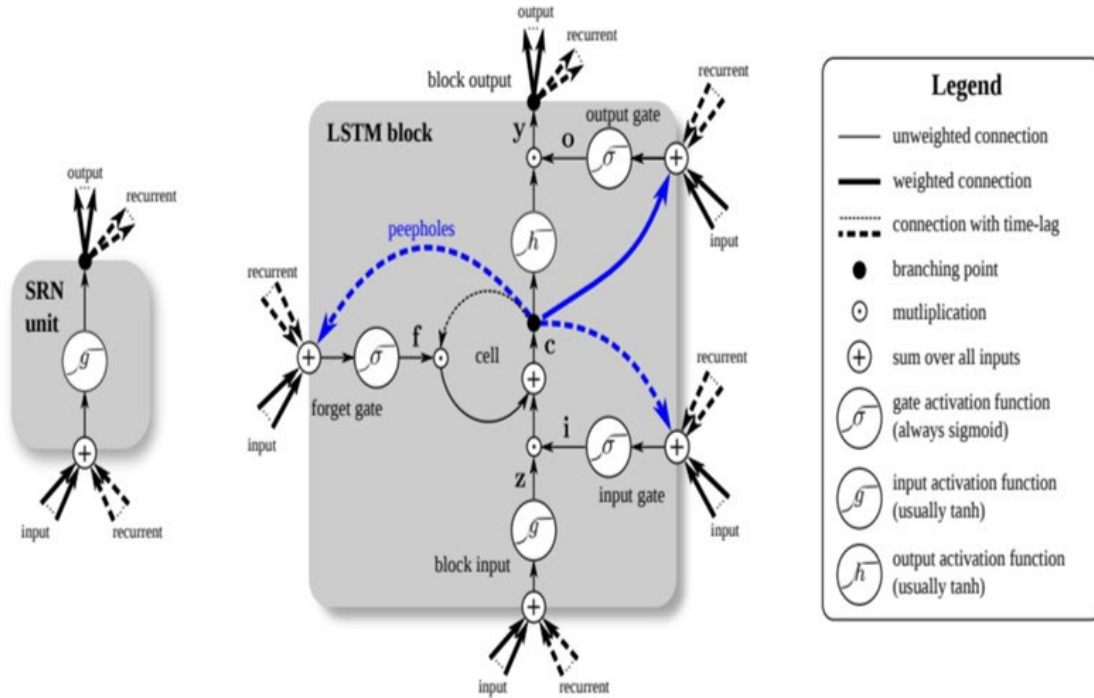


Figure 2.2: Overview Schematic of the Simple Recurrent Network (Greff et al., 2016; Selvin et al., 2017)

The functioning of LSTM can be visualized by understanding the functioning of the three standard step in LSTM. A typical LSTM network comprises multiple memory blocks known as cells. The cell state and the concealed state are the two states that are passed to the following cell. The LSTM can remove or add information to the cell state, which is carefully controlled by structures known as gates. Gates are a method of optionally allowing information to pass through. They are built with a sigmoid neural net layer and pointwise multiplication. The memory blocks are in charge of remembering things, and this memory is manipulated through five main primary mechanisms: the sigmoid layer, tanh layer, forget gate, input gates, update cell state, and output gate. Figure 2.3 shows an overview of the LSTM architecture model (Greff et al., 2016; Selvin et al., 2017).

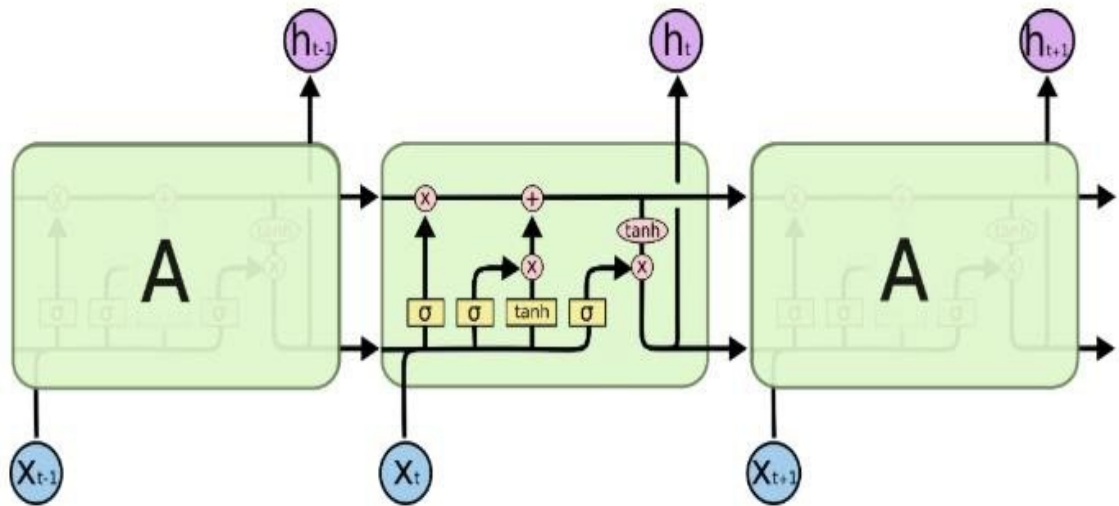


Figure 2.3: Long-Term Short-Term Architecture Model (LSTM) (Gerstner et al., 2014; Greff et al., 2016; Selvin et al., 2017)

2.2.1.1 Sigmoid Layer

The key to LSTMs is the cell state, the horizontal line running, as shown in Figure 2.4. The cell state functions like a conveyor belt. It runs straight down the entire chain, with only some minor linear interactions that allow information to flow along it unchanged.

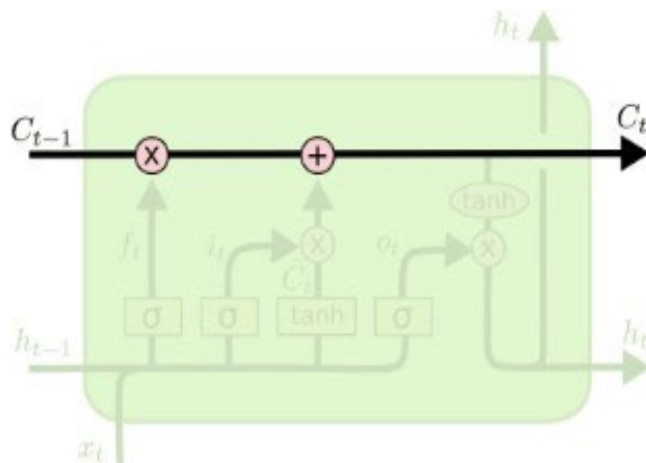


Figure 2.4: Horizontal Line Running Through the Top of LSTM Cell State (Gerstner et al., 2014; Greff et al., 2016; Selvin et al., 2017)

Figure 2.5 shows how the sigmoid layer outputs integers ranging from zero to one, indicating how much each component should be allowed. A value of zero indicates "let nothing through," while a value of one indicates "allow everything through!"

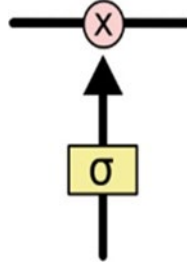


Figure 2.5: Sigmoid Layer in LSTM (Gerstner et al., 2014; Greff et al., 2016; Selvin et al., 2017)

2.2.1.2 Forget Gate

A forget gate is in charge of erasing data from the cell state. The information that is no longer necessary for the LSTM is eliminated by filter multiplication, which is necessary for optimizing the LSTM network's performance. Figure 2.6 shows an overview of the Forget Gate. This gate takes in two inputs; h_{t-1} and x_t , h_{t-1} is the hidden state from the previous cell or the output of the previous cell, and x_t is the input at that particular time step, as shown in Equation 2.1. The sigmoid function outputs a vector, with values ranging from 0 to 1, corresponding to each number in the cell state (Greff et al., 2016; Selvin et al., 2017).

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (2.1)$$

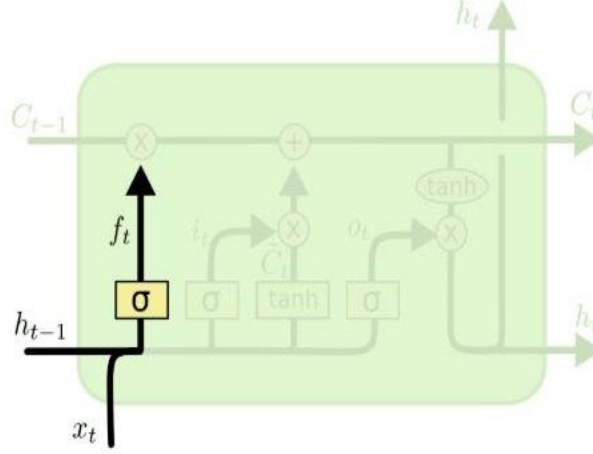


Figure 2.6: LSTM “Forget Gate” (Gerstner et al., 2014; Greff et al., 2016; Selvin et al., 2017)

2.2.1.3 Input Gate and Tanh Layer

The cell state receives information through the input gate, which involves two parts. Figure 2.7 shows an overview of the Input Gate (Greff et al., 2016; Selvin et al., 2017). The "input gate layer," a sigmoid layer, first determines which values to update. Following that, a tanh layer generates a vector of new candidate values. C_t that could be added to the state as shown in Equation 2.2 and Equation 2.3. Combining the sigmoid input gate layer and tanh layer will update the cell state and replace the old value.

$$i_t = \sigma(W_c[h_{t-1}, x_t] + b_i) \quad (2.2)$$

$$C_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \quad (2.3)$$

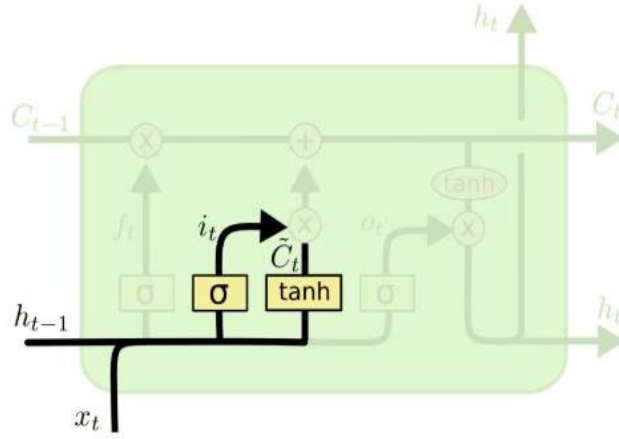


Figure 2.7: LSTM “Input Gate and Tanh Layer” (Gerstner et al., 2014; Greff et al., 2016; Selvin et al., 2017)

2.2.1.4 Update Cell State

This step is to update the old cell state, C_{t-1} , into the new cell state C_t . To activate the forget gate, multiply the previous state by f_c , and the LSTM then work to discard the previous decisions. This approach ensures that LSTM can make accurate predictions based on current inputs without being weighed down by outdated information. Then, adding in i_t C_t will update the new candidate values as shown in Equation 2.4. Figure 2.8 show the architecture of the updated cell state.

$$C_t = f_c * C_{t-1} + i_t * C_t \quad (2.4)$$

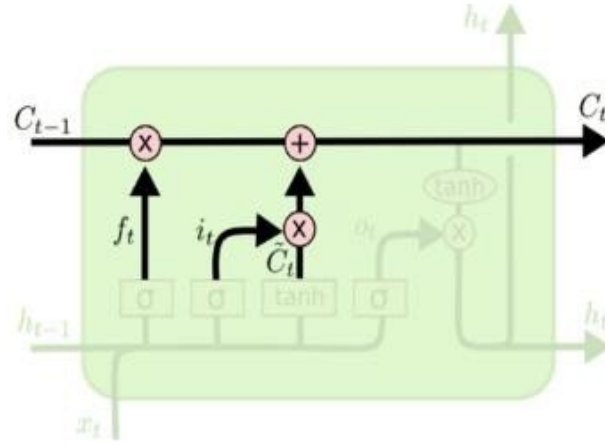


Figure 2.8: LSTM “Update Cell State” (Gerstner et al., 2014; Greff et al., 2016; Selvin et al., 2017)

2.2.1.5 Output Gate

The cell state will determine this output gate. First, as shown in Figure 2.9, run a sigmoid layer to determine which bits of the cell state will be output. Then, in the cell state, apply the tanh function by scaling the values to the range -1 to 1 and multiplying it by the output of the sigmoid gate, which will produce the specific output port as shown in Equation 2.5 and Equation 2.6.

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (2.5)$$

$$h_t = o_t * \tanh(C_t) \quad (2.6)$$

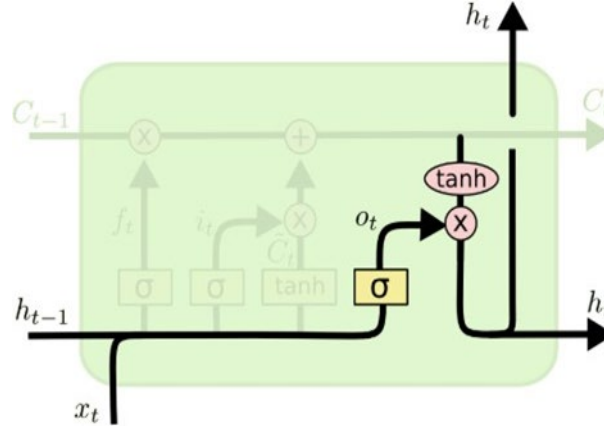


Figure 2.9: LSTM “Output” (Gerstner et al., 2014; Greff et al., 2016; Selvin et al., 2017)

2.3 Spiking Neural Network (SNN)

Spiking neural networks (SNN) are the solution for bridging the gap between computational costs, and SNN can theoretically approximate any function as an ANN (Mass, 1997). Unlike traditional ANN, neurons in SNN communicate with one another through discrete events (spikes) rather than continuous, valued activations. SNN is naturally suited to processing data from AER-based sensors with low redundancy, low latency, and extensive dynamic range, such as DVS (Dynamic Vision Sensor) (Lichtsteiner et al., 2006) and auditory sensors (silicon cochlea) (Liu et al., 2010).

SNNs, part of the third generation of traditional ANNs, convey information through action potentials, or spikes, that send sequences of timely events. A single neuron produces a "spike train," a sequence of action potentials. These events can be viewed as binary occurrences that occur at specific intervals of time, denoted as "on" or "off," depending on whether a spiking event occurs within a time step. The frequency and timing of these binary occurrences can encode information. The rate coding hypothesis suggests that the spiking rate encodes the strength of a stimulus, where the frequency of action potentials increases as

the intensity of the stimulus increases. This works best when the stimulus changes slowly and the observation time is long enough to determine the firing rate.

Johansson and Birznieks (2004) proposed a time-to-first-spike coding approach for encoding tactile afferent information. This approach involves a rapid spike following a reference signal for intense stimulation, while a delayed action potential indicates a weaker input. Meanwhile, Von Der Malsburg (1994) observed the simultaneous firing of neurons in the brain conveying the same concept during object recognition (Gray & Singer, 1989). For fast object recognition, Gautrais and Thorpe (1998) introduced rank-order coding, which prioritizes the relative order of spikes within a group of neurons and disregards the precise timing of spikes.

Several models have been proposed to simulate spiking neural activity. The Hodgkin-Huxley model postulates that the conductance of three channels (chloride, sodium, and potassium) influences axon spike activity (Gerstner et al., 2014; Hodgkin & Huxley, 1952). As mentioned earlier, the model represents spike activity through an electric circuit comprising the channels. There are two main types of SNN models: abstraction threshold models and conductance models. The threshold model uses a fixed threshold voltage value to represent a higher level of abstraction. The neuron will only fire when it reaches this membrane potential. Conductance models can be further divided into the Hodgkin-Huxley and Izhikevich SNN models. On the other hand, the threshold model can be classified as either the Leaky Integrate and Fire (LIF) model or the Spike Response Model (SRM).

2.3.1 Leaky Integrate and Fire (LIF)

The LIF model is a popular threshold-fire model used in SNNs due to its high execution intensity and low computing cost. Its primary circuit consists of a capacitor (C)

connected in parallel with a resistor (R). To reach the post-synaptic neuron, a pulse from a pre-synaptic neuron is filtered through a low-pass RC filter. The diagram in Figure 2.10 for the integrate and fire model provides an understanding of this LIF algorithm. The cell membrane functions like an RC circuit, with input current ($I(t)$) flowing into a neuron and charging the capacitor (C), while simultaneously leaking through the resistance (R) connected to a battery (Gerstner et al., 2014).

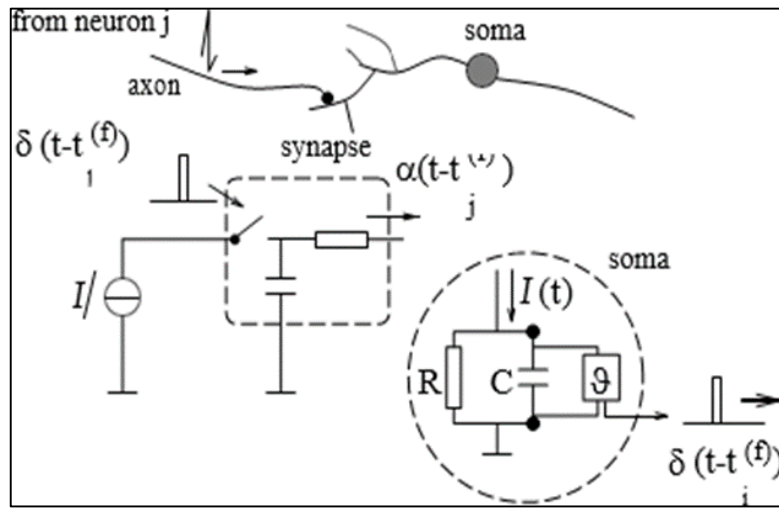


Figure 2.10: Diagram of the LIF (Gerstner et al., 2014)

The model is described by the dynamics of the neuron's membrane potential., $v(t)$ as show in Equation 2.7.

$$C_m = \frac{dv(t)}{dt} I_{leak}(t) + I_s(t) + I_{inj}(t) \quad (2.7)$$

Where C_m is the membrane capacitance, $I_{leak}(t)$ is the current caused to the passive leak of the membrane, $I_s(t)$ is a current describing the effect of synaptic input on the neuron and $I_{inj}(t)$ is an electric current injected into a neuron (by an intracellular electrode).

The leak current of LIF is shown in Equation 2.8.

$$I_{leak}(t) = \frac{C_m}{\tau_m} = [v(t) - V_0] \quad (2.8)$$

Where V_0 is the resting potential, τ_m is the passive membrane time constant, $\tau_m = RC$ to the capacitance and the leak resistance and R_m of the membrane potential.

When the membrane potential reaches the spiking threshold, it generates an action potential that causes neurons to integrate and fire. This is illustrated by Equation 2.9. The spiking mechanism, which is described as a spike when the membrane potential reaches a certain point, formally includes I_{spike} (Gerstner et al., 2014).

$$I_{spike}(t) = \left[C_m \frac{dv(t)}{dt} \right] v = \frac{-1}{V_{th}} (V_{reset} - V_{th}) \delta [v(t) - V_{th}] \quad (2.9)$$

Where V_{th} spike when the membrane potential reaches the spike generating threshold, $V_{reset} (\delta)$ denotes the Dirac Delta function and the membrane potential resumes its evolution according to Equation 2.7.

2.4 Related Work and Discussion

This study used the DSNN algorithm to predict floods in Kuala Baram, Miri. Moreover, Chapter 2 offers a comprehensive summary of various ML and DSNN algorithms previously used for prediction. Chapter 2 will thoroughly examine past study done on the prediction field utilising ML, DL approaches. This review offers a comprehensive analysis of previous literature on related work. Table 2.1 review Deep Learning and Machine related works, Table 2.2 review Machine Learning Application related works, and last Table 2.3 review Hybrid Machine Learning in Flood Prediction related works.

2.4.1 Comprehensive Literature Review of DL and ML in Flood Prediction Related Works

Table 2.1: Summary of DL and ML in Flood Prediction Related Works

No	References	Aim of Study	Models	Performance Metrics	Limitations
1.	(Chen et al., 2023)	An Urban Flooding Prediction Model- Based LSTM Network and Numerical Model	• LSTM • Numerical Model (NM)	• Ponding Depth Error • Depth of Water Accumulation	• No assessment on RMSE and MAE • Lack of involvement of Windows-side technique with LSTM • Noticeable error rate
2.	(Kasyanto et al., 2023)	Application of Multilayer Perceptron (MLP) Method for Streamflow Forecasting (Case Study: Upper Citarum River, Indonesia)	• MLP	• RMSE • NSE • R^2	• Lack of metrics analysis • The model becomes less accurate at longer prediction times
3.	(Yap & Musa, 2023)	Stream Flow Forecasting on Pahang River by Time Series Models, ARMA, ARIMA and SARIMA	• ARMA • ARIMA • SARIMA	• MAPE • Autocorrelation Function (ACF) • Partial Autocorrelation Function (PACF)	• Lack of metrics analysis • To ensure high accuracy, the time series model requires constant updates on data.

Table 2.1 continued

4.	(Ratnasari & Hasin, 2022)	Flood Detection using Backpropagation Neural Network Method	• BP	• RMSE • True Negative (TN) • False Negative (FN) • False Positive (FP) • True Positive (TP)	• Only BP being used for prediction • No comparison with other classical model to validate the experiment
5.	(Balogun & Adebisi, 2021)	A sea Level Prediction using ML and DL Algorithms	• ARIMA • SVR • LSTM	• Heat Map • r • Accuracy • Correlation Pattern	• No forecast model for predicting future rain events • No specifics deployment strategy
6.	(Lin et al., 2020)	Multistep Flood Inundation Forecasts with Resilient Backpropagation Neural Networks: Kulmbach Case Study	• BP • MLP	• RMSE	• Only RMSE being used for analysis the model performance • It is important to differentiate between the accuracy of the forecast for water depth and flood extent.

Table 2.1 continued

7.	(Xiang et al., 2020)	A Rainfall Prediction using LSTM-seq2seq Model	• Lasso Regression • Ridge Regression • Linear regression • Gaussian Processes • Regression • SVR • LSTM-seq2seq	• Nash-Sutcliffe Model • Coefficient correlation (r) • Normalised • RMSE • Statistical Bias	• It proposed LSTM-based seq2seq with different topographies and different watersheds. • Lack of metrics analysis • The model only uses hourly rainfall runoff observation, and rainfall forecast.
8.	(Hu et al., 2020)	A LSTM Model for Streamflow Forecasting of Small River	• LSTM • SVR • MLP	• RMSE • MAE • Coefficient of Determination	• Model tested with outdated data • Significant errors in top-volume forecasts.
9.	(Damavandi et al., 2019)	Streamflow Prediction based on LSTM	• LSTM	• Pearson correlation • Nash–Sutcliffe Model	• No analysis for other metrics • Lack of information on deployment strategy • Prediction process relies on one model, which sustainable for high error rate

Table 2.1 continued

10.	(Zhang et al., 2018)	A Prediction Model for Water Table Depth in Agricultural Areas using LSTM	• LSTM	• RMSE • R^2	• Water depth prediction based on LSTM algorithm. • Performance evaluated using older dataset.
11.	(Widiasari et al., 2018)	A Flood and Water Elevation Prediction Model using Context - based Hydrology LSTM	• LSTM	• MAPE • RMSE	• System relies solely on LSTM algorithm • Unreliable predictions possible • No details on dataset accessibility
12.	(Kratzert et al., 2018)	A Rainfall Prediction Model using LSTM Networks	• LSTM	• Correlation coefficient observed and simulated • Flow Duration Curve • Total Volume Bias • Discharge Variance Bias	• Lack of analysis in terms of metrics • Only LSTM being used for prediction • No involvement of rain sensors to improve model
13.	(Yan et al., 2018)	Urban Flash Flood Forecast using Support Vector Machine and Numerical Simulation	• SVM • NM	• RMSE • Coefficient Correlation • Coefficient Efficiency - TP	• Lack of metrics analysis • More data sampling is needed to validate the model performance.

Table 2.1 continued

14.	(Liu et al., 2017)	A Flood Forecasting Model based on Deep Learning Algorithm via Integrating Stacked Autoencoders with BP Neural Network	• SA • BP • RBF • SVM	• MSE • R^2	• Lack of metrics analysis • Data distribution imbalance
15.	(Shi et al., 2016)	Flood Prediction using Support Vector Machines (SVM)	• SVM	• Coefficient Correlation	• Lack of metrics analysis • Only SVM being used for prediction • Peak flow predictions are not reliable for flood occurrence due to a lack of variables • More variables are needed to improve long-term predictive accuracy
16.	(Phitakwinai et al., 2016)	Multilayer Perceptron with Cuckoo Search in Water Level Prediction for Flood Forecasting	• MLP • Cuckoo Search Algorithm • Blind Test Algorithm	• MAPE • MAE	• Lack of metrics analysis • More variables and machine learning are needed to prove the reliability of the model proposed • Only MLP being used for prediction

Table 2.1 continued

17.	(Keong et al., 2016)	Levenberg-Marquardt Flood Prediction for Sungai Isap Residence	• BP • Levenberg Marquardt Algorithm	• MSE • Batch Ratio	• Lack of metrics analysis • Only MSE and batch ratio (80%: 10%: 10%) and (70%: 15%: 15) being used to analysis the model performance
18.	(Soomlek et al., 2015)	Using BP for Flood Forecasting in PhraNakhon Si Ayutthaya, Thailand	• BP	• MSE • Correlation Coefficient • Model Efficiency Index	• Only BP being used for prediction • There is no clear explanation for training and testing on the data.
19.	(Ruslan et al., 2015)	Flood Prediction Modeling using Improved MLPNN Structure: Case study Kuala Lumpur	• MLP	• RMSE	• Lack of metrics analysis • Only MLP being used for prediction
20.	(Bustami et al., 2007)	Artificial Neural Network for Precipitation and Water Level Predictions of Bedup River	• BP • MLP	• ACC	• Lack of metrics analysis • More variables and ANN are needed to prove the reliability of the model proposed

Chen et al. (2023) developed a new method for predicting urban flooding risk by combining the Long Short-Term Memory (LSTM) neural network model with a hydrological hydrodynamics model. The researchers applied this method to simulate and predict the waterlogging process in the central city of Zhoukou. By integrating the LSTM neural network model with the numerical model of urban flooding accumulation, they were able to accurately predict urban flooding accumulation. The results showed that the ANN model has significant advantages in prediction. Specifically, the LSTM neural network simulation achieved an average error of only 6.34% in accurately predicting the maximum water accumulation depth. Additionally, the prediction speed of the LSTM neural network model was exceptionally fast, making it suitable for emergency flood control. The researchers applied the model to 832 rainfall scenarios lasting 3 hours and generated 20,422 storm flood samples. 90% of these samples were used for training, and the trained results showed an error of only 2.53% compared to the numerical model, with an average error of 5.11% compared to the measured value.

Kasyanto et al. (2023) conducted a study to predict streamflow in the Citaru River at the Dayeuhkolot water level station utilising a Multilayer Perceptron (MLP) artificial neural network. The MLP was trained with rainfall and discharge data collected from thirteen meteorological stations and one hydrological station in Dayeuhkolot for two years, from January 2018 to December 2019. The study evaluated the accuracy and precision of the MLP method by using two distinct input data combinations for multi-hour ahead streamflow forecasting at the Dayeuhkolot Station. The first scenario employed point rainfall data from the thirteen-gauge stations upstream and discharge data recorded at the Dayeuhkolot station as input data, while the second scenario comprised the streamflow at the target station (Dayeuhkolot) and the data on average rainfall at the eight sub-basins. The most optimal

parameters for batch size and epoch values were determined to be 10 and 100, respectively, using the GridSearchCV method. The discharge prediction model was trained and tested one hour ahead. The presented NSE, R², and RMSE values for each data input scenario's prediction results for the next hour of discharge. The models with the highest NSE, R², and lowest RMSE values were considered the best forecasting models. Both scenarios performed well, with a low RMSE value and a high NSE and R² value (greater than 0.9). Scenario 1 was slightly better at predicting discharge one hour ahead than Scenario 2. However, the model's accuracy decreased as the prediction time was extended from 2 to 24 hours. The value of NSE and R² decreased, while the value of RMSE increased, indicating that the model becomes less accurate at longer prediction times.

Yap and Musa (2023) conducted a study demonstrating how time series models can create a short-term river flow forecast model based on historical data from the River Pahang. The study aimed to compare the performance of SARIMA, ARIMA, and ARMA models to determine the best short-term prediction. The monthly streamflow data from January 2010 to September 2019 was used to create the forecast, while data from October to December 2019 were collected for comparison. The study used one input: the streamflow data obtained from the Department of Irrigation and Drainage Malaysia. The study applied three different time series models, ARIMA, ARMA, and SARIMA, to develop the forecast series for two stations, Temerloh and Lubok Paku of Pahang River. The study found that SARIMA (2,0,3) (0,1,0)₁₂ for Temerloh was the best forecasting model, with a MAPE of 18.35%, indicating good modeling accuracy. Additionally, SARIMA (0,0,0) (1,1,2)₁₂ for Lubok Paku demonstrated high accuracy, with a MAPE value of 6.54%. The study found that SARIMA was better at dealing with streamflow seasonality and increased forecasting precision. At the

same time, ARMA models revealed lower MAPE values than ARIMA models due to the over-differencing of the ARIMA model, which lowered its accuracy.

Ratnasari and Hasin (2022) conducted a study to determine the optimal BP neural network architecture for detecting floods. The study involved obtaining a dataset from a specific location, entering it into a database, and ensuring good quality data through data normalization. The study determined the initial weight values, the percentage of training and testing data, and processed the available training data through forward propagation and backpropagation phases until the error rate was lower than the MSE. The classification results showed three categories: not flooded, will flood, and flooded. To obtain a normalization score between 0 and 1, the min-max method was used. After completing the backpropagation neural network design process, testing was performed on the data training process. The optimal architecture or model with the smallest MSE value was then determined to make accurate predictions. The Backpropagation Neural Network method was applied in real-time to predict incoming floods in rivers and watersheds with an accuracy level of 94.3%. The most optimal accuracy for the Backpropagation Neural Network model to predict floods was achieved using the architectural model 5-15-3. This model consisted of 5 neurons in the input layer, 15 in the hidden layer, and 3 in the output layer, with a learning rate of 0.1 and resulted in an RMSE value of $4.65E-11$. These findings provide valuable insights into the use of BP neural networks to accurately predict floods in real-time.

A recent study conducted by Balogun and Adebisi (2021) utilised machine learning and deep learning techniques to predict sea level variation along the coastline of West Peninsular Malaysia. The study examined four different scenarios with various combinations of variables, including sea surface temperature, salinity, density, atmospheric pressure, wind speed, cloud cover, precipitation, and sea level data. The researchers trained ARIMA, SVR,

and LSTM neural network models to assess their performance in predicting sea level changes at different stations. The study found that the LSTM model trained with both ocean and atmospheric variables was the most effective in predicting sea level changes at most stations, except at Pulau Langkawi, where the tide is the most significant factor, and the ARIMA model trained without these variables was the most effective. The study also compared the prediction performance of different models, including univariate ARIMA models, SVR models trained on specific variables, and LSTM models trained on various combinations of variables. The results showed that cloud cover, wind speed, and precipitation consistently correlate with sea level variation across all stations, while other variables have varying correlations.

Lin et al. (2020) conducted a study to predict multi-step flood events for 1–5 hours following their occurrence. To achieve this, the researchers built upon an existing Artificial Neural Network (ANN) previously developed for real-time flood inundation forecasts. Their expanded model boasted a high spatial resolution of 4 meters by 4 meters. The ANN was trained using 120 synthetic flood events and then tested on 60 other synthetic events to verify its forecast performance for lead times of 3, 6, 9, and 12 hours. The model performed well, with over 81% of all grids producing an RMSE below 0.3 meters. The ANN was then applied to three historical flood events to test its ability to make multi-step inundation forecasts. The results showed that the ANN produced highly accurate water depth forecasts for at least 3 hours with over 70% accuracy (RMSE within 0.3 meters) and moderately accurate subsequent forecasts with at least 60% accuracy. This study provides valuable insight into the potential of ANNs to improve flood forecasting and mitigate the associated risks.

The study by Xiang et al. (2020) presented a seq2seq learning model for rainfall-runoff predictions, utilising one layer of LSTM to handle input and output sequences. The

model considers station-based watersheds as a single hydrology unit, assuming that the complex topography remains constant during the study period. This approach is advantageous due to the model's non-linear algorithms and the use of input and output as two time-series sequences. While the model solely predicts the runoff, the seq2seq structure mitigates prediction bias and error by considering the output sequence. The proposed LSTM-based seq2seq model is similar to physical models and significantly enhances modelling accuracy compared to traditional LSTM models. Furthermore, the model applies to various watersheds with differing topography.

Hu et al. (2020) conducted a study that showed that the Long-short-term memory (LSTM) model is a more efficient approach in predicting streamflow in small rivers compared to other models, such as Support Vector Regression (SVR) and Multilayer Perceptron (MLP). LSTM is an upgraded version of the Recurrent Neural Network (RNN), which incorporates a remember-forget system to lessen the probability of parameters vanishing or gradient exploding. However, the original data used in the study had some deficiencies due to the past lack of technology and management, necessitating modifications to enhance accuracy.

Damavandi et al. (2019) conducted a study with the objective of using the Long Short-term Memory (LSTM) network to predict the stream flow of a particular outlet within a watershed. The accuracy of the forecast was evaluated for a United States Geological Survey (USGS) gauge station near Hempstead, located in the Brazos Basin in Texas, which is a poorly gauged region. Data from 2007-2010 was analysed, with antecedent precipitation observations and climate variability indices utilised as potential predictors. The scalability of the model makes it applicable to varying basins with different drainage areas. The performance of the model was assessed using the Pearson correlation and Nash-Sutcliffe

model efficiency coefficients, with an achieved NSE of 0.9542 and 0.8859, respectively, indicating a high level of accuracy in streamflow prediction.

Zhang et al. (2018) conducted a study to develop a precise and efficient method of forecasting water table depth, with the aim of aiding engineers and decision-makers in managing groundwater resources in agricultural areas. To evaluate the proposed LSTM model across the Hetao Irrigation District, water table depth changes were observed throughout the entire area. The model considers the average temperature of five sub-areas, as well as the combined precipitation, water diversion, and evaporation volume of five sub-areas. The study introduces a novel framework comprising one LSTM layer and one fully connected layer for predicting water table depth, deviating from the traditional neural network models typically used in hydrology. The fully connected layer, positioned above the LSTM layer, enhances the model's learning and fitting abilities.

Widiasari et al. (2018) utilised the Long Short-Term Memory (LSTM) algorithm to predict water elevation levels in the upstream area of Samarang, Indonesia. A comparison was made with Multiple Linear Regression, with LSTM outperforming it with a Mean Absolute Percentage Error (MAPE) value of only 3.6%. This indicates that the model's predictions of downstream river water levels have an error percentage of only 3.6% compared to the actual data used for testing, thus demonstrating the greater accuracy of LSTM as a method for predicting water elevation levels.

According to a study by Kratzert et al., (2018), a unique data-driven method employing the Long Short-Term Memory (LSTM) network, a type of recurrent neural network, is introduced. The LSTM is advantageous due to its ability to recognize long-term connections between network inputs and outputs, crucial for modelling storage effects in

snow-influenced catchments. The approach was tested using 241 catchments from the accessible CAMELS dataset and compared the results with the well-known Sacramento Soil Moisture Accounting Model (SAC-SMA) combined with the Snow-17 snow routine. The LSTM was also demonstrated to serve as a regional hydrological model, predicting discharge for various catchments. In the final experiment, the possibility of transferring process understanding gained at the regional level to individual catchments was explored, which significantly improved model performance compared to using an LSTM trained solely on data from individual catchments. This approach outperformed SAC-SMA and Snow-17, highlighting the potential of the LSTM in hydrological modelling applications.

A research study by Yan et al. (2018) suggests the use of support vector machines and numerical simulation to predict urban flash floods. The study developed a framework that combined the SVM model with a well-calibrated numerical model based on MIKE FLOOD to generate real-time forecasts of urban floods. The SVM models were trained using 77 synthetic rainfall events and tested using 34 events. The performance indicators, including the root mean square error (RMSE), mean bias error (MBE), Precise rate (PR), and TPRH were all above 90%, while the coefficient of efficiency (CE) and coefficient of correlation (CC) were above 0.9. The absolute biases were below 0.05 m. This indicated that the SVM models could generate highly accurate urban flood forecasts and were 15 million times faster than the numerical model. The approach allows for the development of real-time urban flood forecast systems at limited monitoring costs.

Shi et al. (2016) proposed a data mining approach for flood forecasting using Support Vector Machines (SVM). The study aimed to predict river flow at a specific time by analysing past rainfall data, focusing on negative lags. The researchers found that rainfall recorded by all gauge stations at time $t-1$ had the strongest correlation (approximately 0.6)

with river flow at time t . SVM showed high accuracy in predicting changes and peak flows in the river over short periods, which could be used to issue timely flood warnings. However, accurate flood predictions beyond 48 hours require additional explanatory variables such as soil moisture and water to improve long-term predictive accuracy.

A study conducted by Phitakwinai et al. (2016) proposed a river flood prediction system that can predict floods 7 hours in advance for the Ping River in Chiang Mai, Thailand. The researchers used the Cuckoo Search (CS) Algorithm to train the weight vector of the MLP to minimise the error between forecasted and actual values. The system uses data collected from upstream and prediction stations as inputs. The study compared the 7-hour-ahead prediction performance of the CS-MLP model with regular multilayer perceptron. The results indicated that the CS-MLP model with a 4-4-2-1 structure outperformed the regular MLP model with the same structure, starting from the 2-hour model. The average mean absolute errors of the best CS-MLP were 8.971 cm and 6.836 cm on the validation set and the blind test data set, respectively.

A study conducted by Keong et al. (2016) proposed the use of Levenberg-Marquardt back-propagation with two different ratios for training, testing, and validation samples. The ratios used were (80%: 10%: 10%) and (70%: 15%: 15%). This model was designed to replicate the thinking process of the human brain and prevent any influence on the predicted judgment. The study collected data on various factors such as temperature, precipitation, dew point, humidity, sea level pressure, visibility, wind, and river level from January 2013 until May 2015. The results were evaluated based on the Mean Square Error (MSE) and Regression (R). The study found that the Levenberg-Marquardt model with an 80% training sample produced better results than the one with a 70% training sample.

A study conducted by Soomlek et al. (2015) introduced water-level forecasting models that can accurately predict water levels at the C.35 gauging station located in the Chao Phraya River, PhraNakhon Si Ayutthaya, Thailand. The model that demonstrated the highest accuracy was further developed into a flood warning application to mitigate the severity of damages caused by natural disasters. The study utilised data collected at five gauging stations in and around PhraNakhon Si Ayutthaya, Thailand, to train and test the water-level forecasting models. The results showed that the most effective forecasting model for predicting the water level of the Chao Phraya River at the C.35 gauging station in PhraNakhon Si Ayutthaya province was the one-day ahead forecasting model when using the 9-7-1 BPNN (EI=0.9012, mean square error (MSE)=0.1139).

A prediction model for flood water levels using Multilayer Perceptron Neural Network was proposed by Ruslan et al. (2015). The study demonstrated that the model's performance significantly improved with an enhanced structure. The dataset utilised to develop the model was separated into three sets: training, validation, and testing samples. The MLPNN model was trained until it reached the performance goal, and its output was compared to the target output data to assess its predictive performance. However, the MLPNN model failed to accurately predict the flood water level at any given segment, resulting in a low Best Fit score of 21.40% and a high RMSE score of 0.7491 meters. Nonetheless, the Improved MLPNN model followed the actual water level pattern trend, necessitating further analysis through graph mapping. The 1-hour flood water level prediction model has been successfully developed and tested in flood-prone areas in Kuala Lumpur, with significant improvement seen through an improved MLPNN structure.

Bustami et al. (2017) study aim to improve water level prediction at Bedup River with estimations made to absent precipitation data using Artificial Neural Network (ANN).

The study aimed to address the missing precipitation data, which had previously hindered the accuracy of water level predictions in Sarawak. To this end, the researchers utilised Artificial Neural Network (ANN) and its Backpropagation properties to estimate missing precipitation and water level data. The ANN model developed in the study achieved an impressive 96.4% accuracy rate in estimating missing precipitation data, which was then used to forecast water levels at the same gauging station with an 85.3% accuracy rate. This significantly improved compared to the 71.1% accuracy rate obtained without estimating missing precipitation data. The study's results highlight the potential of ANN as a valuable tool for hydrologists worldwide in forecasting missing precipitation and water level data.

2.4.2 Comprehensive Literature Review of DL and ML in Flood Application Related Works

Table 2.2: Summary of DL and ML in Flood Application Related Works

No	References	Aim of Study	Models	Performance Metrics	Limitations
1.	(Hansana et al., 2023)	Flood Analysis using Multi-scale Remote Sensing Observations in Laos	• Otsu Method • Google Earth Engine • Digital Elevation Model	• Analytic Hierarchy Process • Correlation coefficient	• Lack of other matrices analysis • Absence of spatial features caused fragmented spots affecting results. • Flood risk map limited by lack of multi-source remote sensing data.
2.	(Faudzi et al., 2023)	IoT based Real-Time Monitoring System of Rainfall and Water Level for Flood Prediction using LSTM Network	• LSTM	• ACC	• It lacks evaluation of other metrics • The researcher based their study analysis solely on Acc. • Only LSTM being used in this study and no comparative done with other method

Table 2.2 continued

3.	(Theilen-Willige & Budiyanto, 2021)	Flooding Hazard Assessment Considering Climate Change in the Coastal Areas of Java-Indonesia Based on Remote Sensing and GIS Data	• Satellite Imageries • Digital Elevation Model (DEM)	• GIS Integrated Evaluation	• It lacks evaluation of other metrics • Only rely on a solid database that has to be updated regularly. • Errors caused by outdated datasets.
4.	(Mousavi et al., 2021)	A Flood Early Detection System Based IoT using ML and DL Techniques	• ANN • LSTM • GRU	• RMSE • Normalised Statistic • Regression	• It limited the study sample. • Lack of other matrices analysis • Do not have rain sensors to observe.
5.	(Sun et al., 2021)	An Efficient Decision Support System for Flood Inundation Management Using Intermittent Remote-Sensing Data	• Decision Support System • SVM • Random Forest • Convolutional Neural Network	• ACC	• The study only uses ACC and does not evaluate other metrics. • Poor quality of optical images caused by bad weather conditions can result in inaccuracies.

Table 2.2 continued

6.	(Dash & Sar, 2020)	Identification and Validation of Potential Flood Hazard Area using GIS -based Multi-Criteria Analysis and Satellite Data-Derived Water Index	• Digital Elevation Model • Synthetic Aperture Radar (SAR) • Remote Sensing	• Decision Making • Multiple Criteria Decision Analysis (MCDA)	• This study has not utilised the sensitivity analysis that was employed in some other studies. • It lacks evaluation of other metrics
7.	(Yariyan et al., 2020)	Flood Susceptibility Mapping using an Improved Analytic Network Process with Statistical Models	• Decision Tree • SVM • Statistical Models • Digital Elevation Model • Decision Making	• ROC • AUC • SEN • ANP	• It lacks evaluation of other metrics
8.	(Bande & Shete, 2017)	Smart flood Disaster Prediction System using IoT & Neural Networks	• Levenberg-Marquardt Algorithm • Nonlinear Autoregressive Exogenous Model	• MSE • Acc	• It lacks evaluation of other metrics • The researcher based their study analysis solely on MSE and Acc.

Table 2.2 continued

9.	(D'Addabbo et al., 2016)	Flooding Monitoring using Synthetic Aperture Radar and Interferometric SAR Data	• Bayesian Network (BN)	• Sensitivity (SEN) • Specificity (SPE) • Probability	• It lacks evaluation of other metrics, such as Accuracy, False Positive Rate, MAPE, etc. • Only uses image datasets • No additional information on rainfall prediction
10.	(Khalaf, et al., 2015)	A Flood Detection and Alert Generation using ML and Sensor Technology	• Random Forest • Bagging Algorithm • Decision Tree • Hyper Pipes Algorithm	• True Positive (TP) • True Negative (TN) • Recall • F-Measure • ROC	• It uses only ML algorithms • No deep learning algorithms • Cannot predict rainfall • Communication relies on GSM technology and requires more infrastructure

Hansana et al. (2023) present a comprehensive workflow utilising the GEE platform for analyzing floods in Laos. The study employs multi-source remote sensing data, automated threshold segmentation algorithms, and comprehensive validation techniques to extract, validate, and analyse flood prone areas. The proposed framework demonstrates robustness in monitoring and analyzing floods at multiple scales. The study utilises Sentinel-1 and Sentinel-2 multi-temporal remote sensing images to calculate the SDWI and NDWI indices accordingly. The Otsu method is employed in automatically calculating the water body segmentation threshold. Decision-level fusion using LULC products, DEM, and its derived data is conducted to accurately extract flood areas. The AHP method is utilised in assessing the flood risk level by integrating crucial data such as rainfall, DEM, and river network. Correlation analysis is conducted to compare the flood detection outcomes with the flood risk map. The proposed method demonstrates exceptional performance, with an average correlation coefficient of 0.7144 over five years, indicating high accuracy in predicting flood risk levels.

Faudzi et al. (2023) proposed a monitoring system for flood forecasting using the LSTM algorithm, which includes a rain gauge and two water level sensors. Historical data indicates that all models are suitable for prediction, but Model 6x4 has marginally better accuracy and validation and has thus been chosen as the preferred model. The accuracy of the results exceeded 95%. For the monitoring system data, Model 10x4 has been selected as highly accurate. Future work could involve the inclusion of rainfall data in prediction and the prediction of different time frames through the accumulation of more data for the system.

The study conducted by Theilen-Willige and Budiyo et al. (2021) attempt to discern flood-prone areas by analyzing their morphometric disposition, mapping the traces of past flooding events from multiple satellite images, and investigating the interplay

between coastal morphology, wind direction, and currents. The study adopts an interdisciplinary approach that employs remote sensing data, geological, geophysical, topographic data, and GIS methods. To construct a GIS database, the researchers utilised satellite images and Digital Elevation Model (DEM) data, which were combined with other geodata and thematic maps. The satellite data utilised in this study encompassed Sentinel 1 - C-Band, Synthetic Aperture Radar (SAR) and optical Sentinel 2 images, Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER), and Landsat visual data (Landsat TM and Landsat 8, the Operational Land Imager-OLI). The impact of climate change on flooding and its escalating toll on society necessitates preparedness and adaptive strategies that rely on an updated and reliable database. By employing Landsat, Sentinel 2, and Sentinel 1 radar images, previous floods can be detected, which is critical for effective flood hazard preparedness and risk estimation, analysis, assessment, and management by local authorities, institutions, and communities. As such, primary data collection is of paramount importance and should be carried out by staff from the municipality in collaboration with local institutions and communities.

Mousavi et al. (2021) proposed an IoT approach utilising LoRaWAN and evaluated three learning models, namely Multilayer Perceptron (MLP), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU), for flood forecasting. The study revealed that Artificial Neural Networks (ANN) failed to learn the model, which is deemed unacceptable. While GRU had lower memory consumption and execution time compared to LSTM. LSTM was more accurate in learning long sequences. The research also discovered that the results of the original data and the data with lost packets were identical. The model's performance was evaluated using three criteria, namely Regression R, Nash-Sutcliffe Efficiency (NSE), and Root Mean Square Error (RMSE), and the regression plot demonstrated that LSTM had

the most precise predictions. The NSE and RMSE values for LSTM were 0.530 and 0.059, respectively. However, the values for GRU were -2.607 and 0.095, which is deemed unacceptable. Based on this case study, it can be concluded that monitoring precipitation upstream and discharge downstream can lead to successful forecasting using LSTM.

Sun et al., (202) conducted a study on the impact of weather conditions during a flood event. Despite the challenging weather conditions that resulted in poor quality optical images, the study successfully developed a dependable decision support system that utilises limited and intermittent visual remote-sensing data. The system effectively tracks floodwater on the ground surface by transforming the flooding process into the numerical solution of the partial differential equation of the boundary function, thereby generating a high-resolution flooding process with both time and space. The study observed that even with limited lower temporal resolution, relatively high-resolution optical RGB imagery can provide information for rapid response mapping of visible floodwater distributions throughout a flood event's life cycle. The decision support system generated in this study could be a significant tool for efficiently extracting visible floodwater from RGB image data and evaluating flood risk through temporal interpolation.

Dash and Sar (2020) conducted a study utilising a GIS-based multi-criteria evaluation framework to identify flood hazard zones in Allahabad district. The study incorporated AHP-based criteria weights and parameters ranking to facilitate decision-making in complex scenarios. The objectives of the study were to generate a flood hazard map to prioritize mitigation measures and to determine flood inundation extent by utilising a multi-spectral satellite image for verification of hazard maps. Within a GIS environment, eight parameters linked to hydro-geomorphological characteristics were formulated and clustered into five groups, and ratings were allocated based on their impact on floods. The

criteria and weights were linearly combined to produce the final hazard map. The findings indicated that approximately 65% of the study area is located in a high to very high hazard zone, with agricultural land being particularly vulnerable to floods.

A study by Yariyan et al. (2020) aimed to improve flood susceptibility mapping in Saqqez City, Kurdistan Province, Iran by combining the Analytic Network Process (ANP) decision-making method with statistical models such as Frequency Ratio (FR), Evidential Belief Function (EBF), and Ordered Weight Average (OWA). Instead of relying solely on expert opinions, the frequency ratio method was used to weigh the criteria in the ANP. The study identified 10 factors that influence flood susceptibility in the area, including slope, rainfall, slope length, topographic wetness index, slope aspect, altitude, curvature, distance from the river, geology, and land use/land cover. A total of 42 flood points were identified in the area, with 70% used for modeling and the remaining 30% used for validation. Results were evaluated using the Receiver Operating Characteristic (ROC) curve, which showed that the ANP-EBF hybrid model performed the best, with 95.1% efficiency, compared to ANP-FR (91%) and ANP-OWA (89.6%).

Bande and Shete (2017) proposed a framework for flood monitoring and prediction that utilises artificial neural networks and IoT technology. The system employs two algorithms to assess the accuracy of the network's output, measuring the mean square error (MSE). The first algorithm provides adequate rainfall prediction, while the second algorithm can predict both rainfall and flooding with an accuracy of approximately 88%. The study identified the Levenberg-Marquardt training algorithm with the NARX network as producing the best results, offering real-time flood prediction with step-ahead alerts. The system utilises ThingTweet to alert at-risk communities in simple language via the internet. By integrating advanced technologies such as IoT, machine learning, big data, predictive

analytics, social media, and mobility, forward-thinking nations can effectively manage emergencies and disasters.

D'Addabo et al. (2016) conducted a study proposing a statistical model that employs Bayesian networks to detect floods by fusing information and data. Bayesian networks are probabilistic graphical models that utilised graph structures to represent the dependence assumptions among the variables involved. The study used a data set that acquired during winter which primarily covered agricultural areas unaffected by agricultural practices during this time of the year, and thus did not exhibit any changes due to the phenological cycle of their cultures. The Bayesian framework combined the data by assigning a priori flood probabilities to individual cluster centroids and incorporating ancillary information, such as the distance from the river course and a geomorphic flooding index (GFI). Experimental results showed that the BN-based data fusion approach has significant capabilities in identifying large areas affected by floods, even in the presence of coherence classes corresponding to different land cover types that respond differently to the water and inundation evolution. This approach mitigates false alarms and can accurately identify flooded areas in events characterized by complex land cover ground conditions and time evolution.

Khalaf et al. (2015) proposed the implementation of a GPS module to track equipment in various flood-prone areas, which can enhance the early warning and forecasting system for floods. Sensor technology is commonly employed to monitor water levels in different locations, and the flood detection system is designed to promptly alert local authorities. SMS and web-based public networks are effective means of notifying flood victims within a specific area. The Random Forest algorithm proves to be highly accurate, with a 99.5% accuracy rate, surpassing the Bagging algorithm's 97.7% accuracy rate.

However, the HyperPipes algorithm exhibits the lowest accuracy percentage, achieving only 89.8%. The primary objective of utilising GSM technology is to deliver real-time SMS service-based water level severity alerts derived from water sensor data

2.4.3 Comprehensive Literature Review of Hybrid DL and ML in Flood Prediction Related Works

Table 2.3: Summary of Hybrid DL and ML in Flood Prediction Related Works

No	References	Aim of Study	Models	Performance Metrics	Limitations
1.	(Di Nunno et al., 2023)	Short-Term Forecasts of Streamflow in the UK based on a Novel Hybrid Artificial Intelligence Algorithm	• MLP-RF Model • NARX models	• MAPE	• It lacks evaluation of other metrics • Require more variables and performance measurement for software and hardware integration
2.	(Sahoo & Ghose, 2022)	Application of Hybrid Support Vector Machine Model for Streamflow Prediction in Barak valley, India	• SVM • Whale Optimisation Algorithm (WOA)	• RMSE • MAE • NSE	• It lacks evaluation of other metrics • Future integration of software and hardware requires additional variables and performance measurements.

Table 2.3 continued

3.	(Azad et al., 2022)	Water Level Prediction through Hybrid SARIMA and ANN Models Based on Time Series Analysis: Red Hills Reservoir Case Study	• SARIMA • ANN	• MAE • MAPE • RMSE • R²	• Lacks evaluation of other metrics • Only one input variables are used in the predictive model
4.	(Moishin et al., 2021)	Designing Deep-Based Learning Flood Forecast Model With ConvLSTM Hybrid Algorithm	• LSTM • CNN	• MAE • MSE • RMSE • Legate-McCabe Efficiency Index (LME)	• Lacks evaluation of other metrics • Only two input variables are used in the predictive model
5.	(Shuofeng et al., 2021)	LSTM based Hybrid Method for Basin Water Level Prediction by using Precipitation Data	• LSTM • MLP • Hybrid LSTM-MLP Model	• RMSE	• Lacks evaluation of other metrics. • Only one input variable is used in the predictive model.

Table 2.3 continued

6.	(Liu et al., 2020)	The Applicability of LSTM-KNN model for Real-Time Flood Forecasting in different Climate Zones in China, Water	• LSTM • KNN	• RMSE • NSE • R^2 • Validation Epoch (VE)	• It lacks evaluation of other metrics • LSTM internal variables cannot be directly interpreted or reflect the internal dynamics of the catchment, which is a challenge that impedes its complete application
7.	(Anupam & Pani, 2020)	Flood Forecasting using a Hybrid Extreme Learning Machine-Particle Swarm Optimization Algorithm (ELM-PSO) Model	• Extreme Learning Machine (ELM) • Particle Swarm Optimization Algorithm (PSO)	• MSE • R^2	• It lacks evaluation of other metrics • Need constant data updates • Require more variables for software and hardware integration

Table 2.3 continued

8.	(Yoo et al., 2020)	Data Driven Water Surface Elevation Forecasting Model with Hybrid Activation Function-A Case Study For Hangang River, South Korea	• Hybrid Activation Function with LSTM	• RMSE • NSE • PE • Acc • Precision • Recall	• Accuracy decreases with longer lead times • Less improvement in the accuracy of the virtual flood prediction model.
9.	(Moeeni & Bonakdari, 2017)	Forecasting Monthly Inflow with Extreme Seasonal Variation using the Hybrid SARIMA-ANN Model	• SARIMA • ANN	• RMSE • Coefficient Correlation	• Lacks evaluation of other metrics. • Only one input variables are used in the predictive model
10.	(Zhong et al., 2017)	A Hybrid Model for Water Level Forecasting: A case Study of Wuhan station	• ANN • Kalman Filter - Hybrid KF-ANN Model	• RMSE • R^2	• Lacks evaluation of other metrics • Further research should be conducted to examine the strategies for integrating the Kalman filtering algorithm in more detail.

A study conducted by Di Nunno et al. (2023) suggests a hybrid model for predicting river flows. The proposed model, MLP-RF (combining Multilayer Perceptron and Random Forest algorithms), was tested in the Thames River in southern England and produced accurate predictions for both regular and high flows. However, the accuracy of forecasts decreased for smaller basins, such as the Ribble River in Northern England, significantly as the forecast horizon increased. This was observed in the NARX and MLP-RF models. To improve the accuracy, the proposed model stacked the NARX, RF, and MLP algorithms and refined the hyperparameters using Bayesian Optimisation. This new model, NARX-MLP-RF, showed less accuracy reduction as the forecasting horizon increased for regular and extreme streamflow compared to the NARX and MLP-RF models. Additionally, the study found a significant correlation between the increase in Mean Absolute Percentage Error (MAPE) and the CV of the flow time series when increasing the forecast horizon from 1 to 7 days. MLP-RF had a MAPE variation of less than 10%. The accurate predictions made with the NARX-MLP-RF model can be a valuable tool for managing risks associated with frequent floods and for short-term water management decision-making.

Sahoo and Ghose (2022) proposed a novel approach to forecasting streamflow at the Badarpur Ghat gauging station in the Barak River basin. Their proposed model employs a hybrid support vector machine (SVM) and Whale Optimisation Algorithm (WOA) to enhance the accuracy of streamflow forecasting. The performance of the SVM-WOA model was compared with a conventional SVM model. The testing phase of the study revealed that the NSE coefficients for the SVM-WOA model ranged from 0.9764 to 0.9863, while the conventional SVM model showed NSE values ranging from 0.9207 to 0.9302. The SVM4 and SVM-WOA4 models demonstrated the best performance, with SVM4 achieving an RMSE of 19.2213, NSE of 0.9536, and MAE of 9.973, while SVM-WOA4 scored an RMSE

of 10.534, NSE of 0.9863, and MAE of 5.028. Accurate streamflow forecasting is crucial in water resource management and river construction. The integration of SVM with WOA has proven to be an effective means of improving the precision of classical SVM techniques and achieving higher precision in predicting streamflow than the SVM-based model. Future research should explore other intelligent systems for Qflow forecasting in the Barak River basin.

A study by Azad et al. (2022) has introduced a new method for accurately predicting Reservoir Water Level (RWL) using a Hybrid SARIMA-ANN model. The initial SARIMA model, SARIMA (0, 0, 1) (0, 3, 2), was found to have residuals that did not meet the autocorrelation requirement. To address this limitation, the researchers developed a new SARIMA-ANN hybrid time series model, which was more effective. The study used monthly RWL data from January 2004 to November 2020 to develop and test the models. The performance of each model was evaluated using various model assessment criteria, and the results showed that the SARIMA-ANN hybrid model outperformed all other models in predicting reservoir RWL. Therefore, the study concludes that the SARIMA-ANN hybrid model is reliable and accurate for predicting reservoir water levels.

A recent study by Moishin et al. (2021) has developed a new flood forecasting model using a hybrid deep learning algorithm called ConvLSTM. This model integrates the advantages of both Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) networks to predict future flood events. The study uses a Flood Index, derived from precipitation data, to monitor and determine the severity, duration, and intensity of any potential flood situation. The newly designed predictive model uses lagged IF, combined with real-time and antecedent rainfall data, to forecast the following daily IF value. The proposed ConvLSTM model was validated against nine different rainfall datasets in Fiji, a

region that faces frequent flood-driven devastation. The results showed that the ConvLSTM-based flood model outperformed all benchmark methods at 1 day, 3 day, 7 day, and 14 day forecast horizons. For example, the Root Mean Squared Error (RMSE) for the study sites was 0.101, 0.150, 0.211, and 0.279 for the four forecasted periods, respectively, using the ConvLSTM model. The next best model had RMSE values of 0.105, 0.154, 0.213, and 0.282 for the same forecast horizons. The Legate-McCabe Efficiency Index (LME) was also used to evaluate the difference in model performance for individual stations. The results showed that the LME index was 0.939, 0.898, 0.832, and 0.726 for the four forecast horizons, respectively. These findings demonstrate the practical utility of ConvLSTM in accurately forecasting IF and its potential use in disaster management and risk mitigation during extreme weather events.

Shuofeng et al. (2021) conducted a study proposing a hybrid model of LSTM-MLP and comparing it with the LSTM and MLP models. The researcher's study aims to introduce a universal and simple prediction model based solely on precipitation data, known as the Long Short-Term Memory (LSTM)-based hybrid model, and assess its accuracy. The researcher recommends utilising a hybrid model based on LSTM to predict small rivers with only precipitation data, as it can predict any precipitation pattern and generate meaningful output even after rainfall ceases. Moreover, the LSTM-based hybrid model demonstrates significantly improved accuracy compared to statistical models, owing to the added constraint of the tank model. The researcher evaluates the prediction accuracy of each model for one to six hours using RMSE analysis. The growth value of the LSTM and MLP models is approximately 0.3, indicating that the LSTM-based hybrid model produces more dependable long-term prediction outcomes. Additionally, the LSTM-based hybrid model's RMSE is lower than the LSTM and MLP models in every prediction time point, even in the

6-hour prediction, where the proposed model's RMSE is lower than the 1-hour RMSE of the other two models.

Liu et al. (2020) conducted a study on the applicability of the LSTM-KNN Model for Real-Time Flood Forecasting in different climate zones in China. The researchers conducted a comparative analysis of this model with the RNN artificial intelligence technique. They developed a conceptual-based XAJ model to identify similarities between the conceptual and data driven models. Furthermore, the researchers proposed a coupled LSTM-KNN model to determine its potential for real-time forecasting. The experiment demonstrated that the LSTM model with internal memory could learn and store long-term dependencies of the input-output relationship in different climates. Additionally, the LSTM model produced comparable results to the conceptual XAJ model, indicating its superior robustness to the simple RNN model. The coupled model with the KNN algorithm demonstrated increased accuracy in predicting runoff in three catchments compared to the LSTM model. Hence, it is a more suitable method for enhancing the accuracy of real-time runoff forecasting.

Anupam and Pani (2020) developed a new flood forecasting method using an ELM-PSO model. This model utilises data from previous days, including mean daily gauge heights, mean daily rainfall, and mean daily river discharge values, to predict mean gauge heights up to 4 days in advance. The ELM-PSO model optimizes the number of units in the ELM to achieve the highest coefficient of determination. Data from four decades at two flood-forecasting stations in Odisha were used to create models for mean gauge height prediction. These models were then evaluated using tenfold cross-validation, with mean-squared error and coefficient of determination as evaluation parameters. Results showed promising accuracy, with the Jenapur station having an MSE of 0.14 and R-squared of 0.85

for 1-day-in-advance predictions, and Anandpur having an MSE of 0.23 and R-squared of 0.75.

Yoo et al. (2020) conducted a study to evaluate the effectiveness of a forecasting model that employed a hybrid activation function. The function was developed to overcome the limitations of single activation functions, which tended to underestimate flood management results in the Hangang River when water surface elevations were above EL: 3.9 m. By utilising the hybrid activation function, the model's range of applicability was reduced, but it was able to provide accurate forecasts at high water surface elevations. The forecasting accuracy was deemed acceptable for up to 3 hours, as determined by root mean square error (RMSE) and Nash-Sutcliffe efficiency (NSE), because the partial autocorrelation coefficient for water surface elevation at the Hangang River was below 95%. However, the model's performance improvement for virtual flood forecasting with the hybrid activation function was not conclusive due to the limited sample size of flood events. Nevertheless, in comparison to other single activation functions, the model utilising the hybrid activation function exhibited superior accuracy, precision, and recall when performing virtual flood forecasting, as evidenced by the results of the water surface elevation comparison.

Moeeni and Bonakdari (2017) proposed a novel approach for flood forecasting using a hybrid extreme learning machine-particle swarm optimization algorithm (ELM-PSO) model. The approach involves utilising the mean daily gauge heights, mean daily rainfall, and mean daily river discharge values of prior days to forecast the mean gauge heights up to four days in advance. The input features are used for an extreme learning machine (ELM) regression model, and the particle swarm optimization algorithm (PSO) was used to optimize the number of units in the ELM to get the highest coefficient of determination. The gauge,

rainfall, and discharge data of four decades from the Jenapur flood-forecasting station (Brahmani River, Odisha) and the Anandpur station (Baitarani River, Odisha) were employed to create models for mean gauge height prediction. These models were then cross-validated using tenfold cross-validation, with mean-squared error (MSE) and the coefficient of determination (R-squared) as parameters for evaluating the models. The models exhibited promising results, with the 1-day-in-advance model having MSE 0.14, R-squared 0.85 for Jenapur, MSE 0.23, and R-squared 0.75 for Anandpur.

According to a study conducted by Zhong et al. (2017), a hybrid model of Artificial Neural Network and Kalman filtering was proposed to predict daily water levels at Wuhan station. The dataset was split into training (80%) and validating (20%) to forecast daily water levels in the Yangtze River, which is essential for marine traffic safety. To address the complexity of the hydrologic time series, a short-term water level forecasting model based on the ANN model was proposed and integrated with Kalman filtering to achieve better forecasting accuracy. The results of the study showed that the hybrid model provided more reliable predictions of daily water levels compared to the ANN model, except for one case in 2015. However, more detailed investigations on incorporating the Kalman filtering algorithm should be conducted. Both natural and artificial factors could affect the temporal and spatial variations of daily water levels in the middle reaches of the Yangtze River. Overall, the study provides valuable insights into applying the hybrid model for daily water level forecasting.

2.5 Summary

This chapter provides an extensive review of the existing literature on flood prediction and associated algorithms, with a particular focus on Deep Learning (DL) and Machine Learning (ML) techniques. The review encompasses prominent algorithms such as Long Short-Term Memory (LSTM), Leaky Integrate and Fire (LIF), and Spiking Neural Network (SNN), as well as the application of DL and ML in flood related works. Additionally, the chapter explores the integration of hybrid approaches in flood related studies. Moving forward, Chapter 3 elaborates on the methodology utilised for the DSNN algorithm in flood prediction

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Overview

This chapter thoroughly examines the methodologies utilised in prior research studies for flood prediction and the techniques implemented in this investigation. This study's primary focus is applying the DSNN algorithm for flood prediction. The rainfall dataset was obtained from the Sarawak Department of Irrigation and Drainage (DID), and data sets were generated specifically for neural network training, validation, and testing. Multiple ML models, including LSTM, BP, MLP, SARIMA, and SVM, were chosen for training and validation in this study. Subsequently, their performance was evaluated and analysed by implementing performance measurement analysis.

3.2 General Research Framework

A new hybridisation, a DSNN algorithm for flood prediction, has been developed. The study presents a comprehensive framework for comparing multiple Machine Learning (ML) models and explains the training process for this approach. The researchers analysed the effectiveness of the DSNN algorithm using performance measurements and data sets from DID. Figure 3.1 illustrates the framework utilised in the study. The findings demonstrate the potential of this methodology in flood prediction, with significant implications for disaster management and prevention.

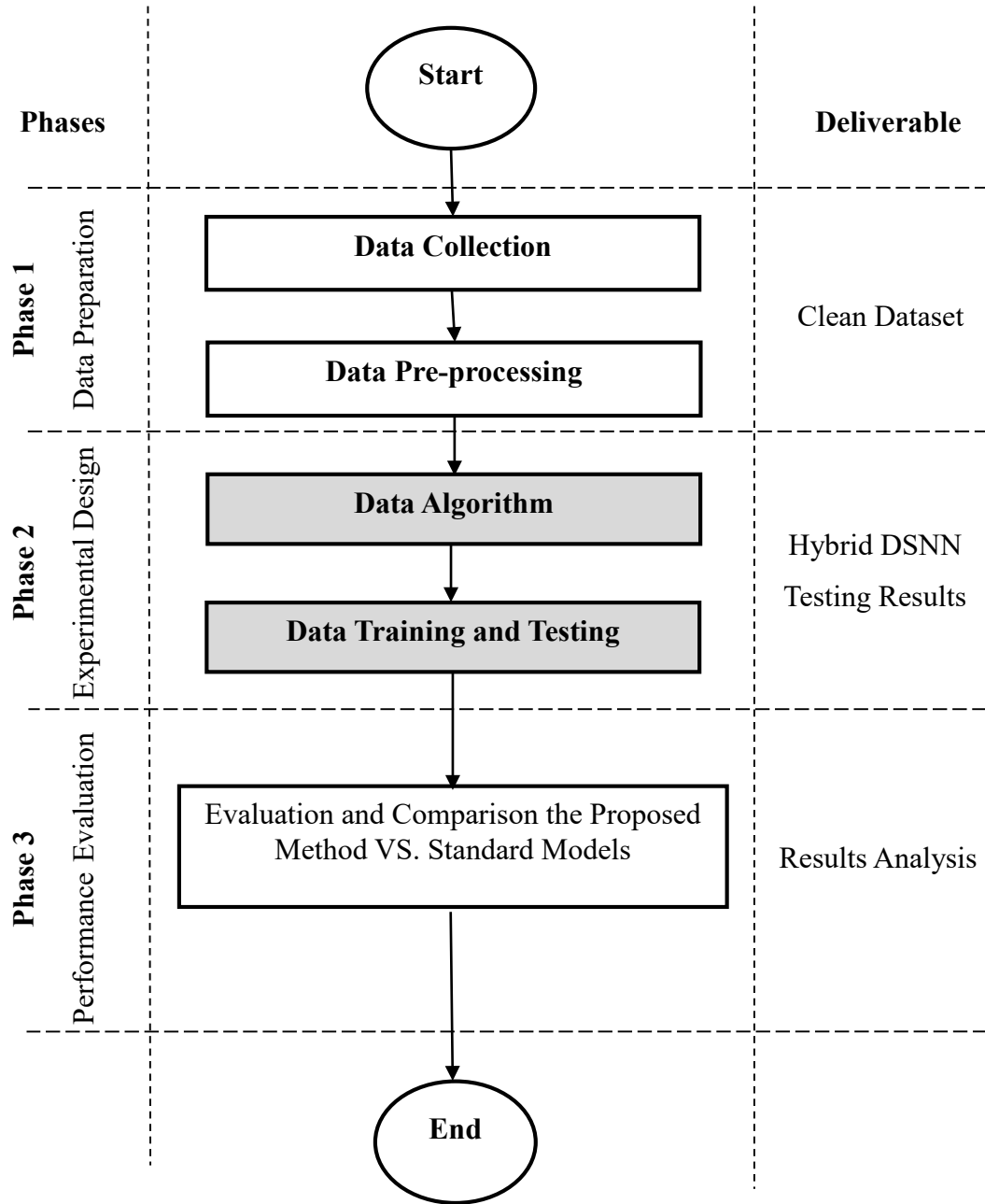


Figure 3.1: Framework of the Study

3.2.1 Phase 1: Data Preparation

The section can be split into two parts: data collection and data pre-processing.

The following sections will describe each step in detail.

3.2.1.1 Phase 1: Data Collection

The area of study is Baram River, Miri, Sarawak. The DID of Sarawak has provided the necessary data, and the Economic Planning Unit of Sarawak has approved the study, which covers 30 years and encompasses 35 basins. The main focus of the study is on rainfall data, and Table 3.1 displays the basin stations for rainfall, Kuala Baram, Miri, Sarawak.

Table 3.1: Rainfall of Basin Stations Kuala Baram, Miri, Sarawak

Types and Unit of Data	Station No	Name of Station	Period of Data
Rainfall Datasets	4440060	Miri DID Barrack	(1989-2019) 30 years Datasets
	3152011	Lio Matu	
	3347003	Long Akah	
	3541033	Long Jegan	
	3744009	Long Lama	
	3945017	Long Panai	
	3842034	Long Teru	
	4143004	Marudi	
	3940036	Beluru	
	4339005	Miri Airport	
	4339001	Miri MCC	
	4440001	Lutong	
	3050015	Long Moh	
	2949001	Long Jee	
	3048026	Long Anap	
	3345029	Long Naha'ah	
	3243071	Long Aton	
	3342032	Long Subing	
	3441001	Long Batan	
	3547001	Long Luteng	
	3444018	Long Pilah	

Table 3.2 continued

	4442001	Tanjong Jaye	
	4043059	Benawa	
	3847035	Long Atip	
	3950020	Long Seridan	
	3752001	Pa Tik	
	3754007	Bario	
	3451028	Long Lellang	
	4151017	Long Napir	
	3946001	Long Terawan	
	4047001	Long Pala	
	4049001	Mulu	
	4139064	Kebuloh	
	4239001	Bukit Lambir Micro	
	4241001	Bakong	

3.2.1.2 Data Pre-Processing

Before classification, the data sets were processed and cleaned to improve DSNN structure and accuracy. Due to their incompetence and unreliability, real-world data sets frequently contain noise. The following two subsections of this study explain data normalisation and data division.

i. Data Normalisation

The data is pre-processed by using a normalisation method, which involves scaling or mapping techniques in problem statements (Al Shalabi et al., 2006; Borkin et al., 2019; Sulaiman & Wahab, 2018). Normalisation is crucial for predicting purposes, as it produces a new output by normalising existing data. This study employs the normalisation technique to pre-process flood data, using Equation 3.1.

$$Z_{normalised} = \frac{Z_0 - Z_{min}}{Z_{max} - Z_{min}} \quad (3.1)$$

Where Z_0 is the largest value of y (before normalisation), $Z_{normalised}$ is the new value Z (after normalisation), Z_{max} is the maximum value of Z in the sample data and Z_{min} is the minimum value of Z in the sample data.

ii. Data Validation

Effective problem-solving requires accurate representation of data, which can be achieved by dividing it into training and testing sets. The training set is utilised to train the data while the testing set evaluates the generalisation performance of Deep Neural Networks (DSNN). It is important to note that different sections of the same data set can vary in significance. Two primary validation methods are hold-out validation and K-fold cross-validation (Moreno-Torres et al., 2012; Pal & Patel, 2020; Yadav & Shukla, 2016). In this case, K-fold cross-validation was employed to verify the algorithm's ability to accurately estimate and detect the overfitting problem. The validation score was determined by considering the average results of the operations, while the true error rate was calculated using the average error rate, as demonstrated by Equation 3.2.

$$S = \frac{1}{K} \sum_{i=1}^K S_i \quad (3.2)$$

Where K is the fold number of data set and S_i is the mean square error for each fold. Lastly, the effectiveness of the proposed method is characterised as a completely objective measurement of accuracy with validation data. The testing section, which provides an impartial approximation of generalisation accuracy, must be used. The validation operational process is as follows:

- Divide the data set into two parts: training and testing.
- Determine the features and structure of the training.
- Train the suggested approach using a portion of the training.
- Repeat steps 2 and 3 with various architectures and training parameters.
- Choose the best model and train it with data from the training section.
- Using the testing section, evaluate the final recommended approach.

The provided data sets were pre-processed and prepared for training, validation, and testing of the DSNN algorithm. Each data set was randomly divided into training data (80), validation data (10) and testing data (10) using cross validation. Four batch ratios were utilised in this study: 80% training, 10% validation, and 10% testing; 70% training, 15% validation, and 15% testing; 60% training, 20% validation, and 20% testing; and 50% training, 25% validation, and 25% testing. The DSNN algorithm was trained and implemented using Python.

3.2.2 Phase 2: Experimental Design

This section explores the approach utilised in a study and compares it with other algorithms. The researchers analysed multiple degrees of the DSNN algorithm to determine the most appropriate hybrid algorithm for the data. To assess the efficiency of the DSNN algorithm, several experiments conducted on 35 datasets using K-fold cross-validation. This involved dividing the data into k subsets of equal size and designing the framework for K iterations. In each cycle, one subset served as the testing dataset, while the rest were used for training. All proposed methods were executed on a PC with an Intel(R) Core(TM) I5-3337U processor, 12GB of installed memory (RAM), and Windows 10 for the data sets. The subsequent section will provide more comprehensive information.

3.2.2.1 Hybridisation of SNN with LSTM (DSNN)

LSTM is a type of dynamic neural network state that is used to train Recurrent Neural Networks (RNN). Recurrent networks have the ability to use feedback connections to store recent input events as short-term storage activations, rather than relying on long-term memory that slowly takes the form of weight changes. These feedback connections can be particularly useful in various applications such as predicting, musical composition, speech processing, and non-Markovian control. LSTM has been successfully applied to a wide range of learning issues, including those that are large and complex. In fact, many studies have already been conducted using LSTM for flood prediction purposes, and the superior advantages of this approach have been demonstrated in numerous research studies. The LSTM consists of three different gates: input, forget, and exit, as well as a block input, a single cell, output activation, and peephole connections. The block output is repeatedly connected to the block input and gates. To understand how LSTM works, it is helpful to consider the three standard steps that are involved. Essentially, a typical LSTM network is

made up of multiple memory blocks that are called cells. These cells are used to store two important states: the cell state and the hidden state. The LSTM can delete or add information to the cell state, and this process is strictly managed by structures known as gates. These gates are essentially a mechanism for passing information selectively, and they consist of a sigmoid neural network layer that is spot propagated.

On the other hand, SNNs proved to be the solution for bridging the gap between computational and theoretical. SNNs communicate with each other via discrete events (spikes) instead of continuous-valued activations. Such a system is updated asynchronously as the event arrives, thus reducing the number of operations required at each time step. The LIF model has been widely used due to its intensity of firing rate execution. LIF can shorten the time of prediction and give better accuracy. LIF shows a formal spiking neuron model. Therefore, this study will adopt the LIF model threshold and merge it into the LSTM model; the thresholds classify the flood datasets.

Figure 3.2 briefly describes the simple schematic representation of the DSNN proposed method. SNN are gaining momentum in resolving complicated time issues. Generally, SNN are computationally robust and efficient. The LIF model threshold will integrate into the LSTM model, where the thresholds play a significant role in the LSTM process by classifying the flood data, returning, and estimating the accuracy of where the threshold is dynamically learning from the datasets, and determining the appropriate threshold from the start of the action. LIF can shorten the time of prediction and give better accuracy. Figure 3.3 illustrates the Pseudo-code of Hybrid DSNN Algorithm.

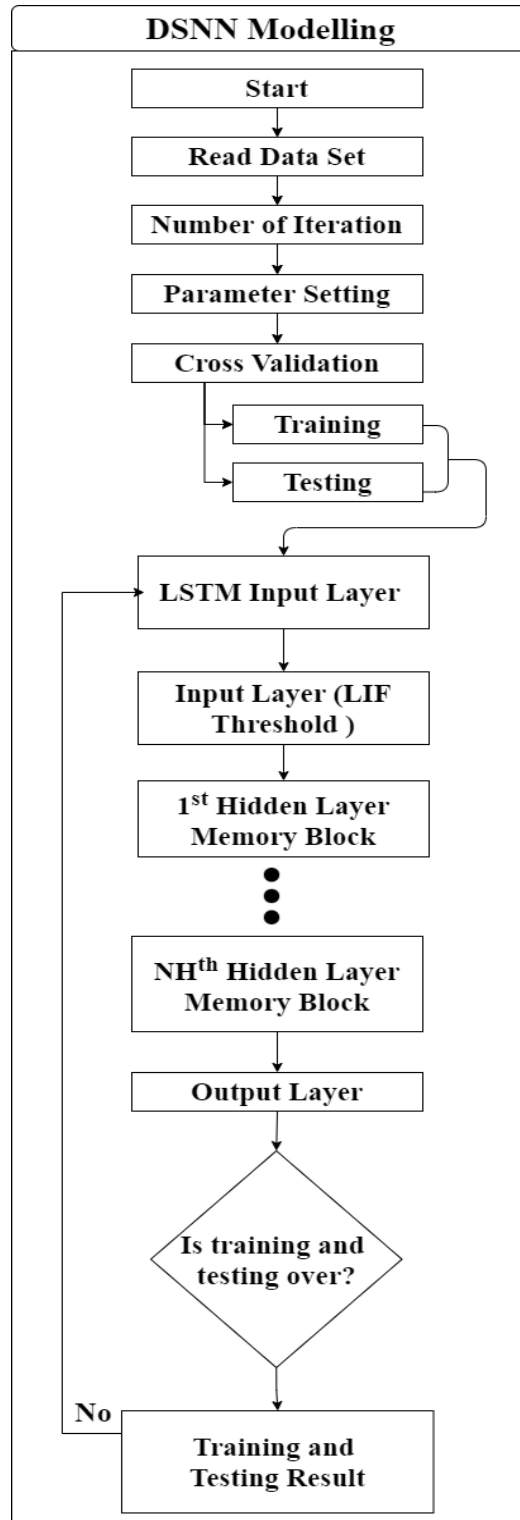


Figure 3.2 Simple Schematic Representation of DSNM Proposed Method

- 1. Begin**
2. Import the required libraries
3. Set a random seed for reproducibility.
4. Load the dataset
5. Split the dataset into training and testing sets
6. Convert the dataset to PyTorch tensors
7. Define hyperparameters: input_size, hidden_size, output_size, learning_rate, and epochs
8. Define a class SpikingLSTM that inherits from the nn.Module class.
9. Inside the class SpikingLSTM, define the constructor function that takes input_size, hidden_size, and output_size as parameters
Initialize the LSTM, spike, and fully connected layers using the nn.LSTM, nn.Sequential, and nn.Linear classes respectively.
10. Inside the class SpikingLSTM, define the forward function that takes x as input
Perform an LSTM operation on x
Spike the output using the spike layer
Apply the fully connected layer to the spiking output
11. Create an instance of the SpikingLSTM class with the specified hyperparameters
12. Create an instance of the Adam optimizer with the model parameters and learning rate
13. Train the model for the specified number of epochs by iterating over the range of epochs
Zero out the gradients of the optimizer.
Obtain the model output on the training data.
Calculate the cross-entropy loss between the model output and the training labels.
Backpropagate the loss through the model.
Update the optimizer parameters.
14. Evaluate the model on the testing set
Disable gradient tracking for evaluation.
Obtain the model output on the testing data.
Predict the labels by taking the argmax of the output.
Calculate the accuracy, sensitivity, recall, and specificity of the model on the testing data.
- 15. End**

Figure 3.3 Pseudo-code of Hybrid DSNN

Integrating SNN with LSTM entails merging the two models to leverage their respective strengths for data analysis and prediction. The SNN utilises spiking neurons to replicate the actions of biological neurons, while the LSTM is a RNN capable of processing time-series data by retaining information from past inputs.

To hybridise these models, the LSTM layer is the input layer for the spiking neural network, a technique that involves evaluating the output of the LSTM layer using spiking neurons. The Python library pandas can be employed to import the dataset into an Excel file. The data can then be imported into a DataFrame object and saved in an Excel file using the `to_excel()` method. Furthermore, the scikit-learn library can be utilised to incorporate performance metrics such as Sensitivity, Recall, and Specificity into the code. This library provides functions that can calculate these metrics. The code initially imports the dataset from an Excel file using the Pandas library and subsequently splits the data into training and testing sets using the `train_test_split` function from scikit-learn. Finally, the data is transformed into PyTorch tensors.

The SpikingLSTM class was created in PyTorch by inheriting from the `nn.Module` class, and essential hyperparameters such as input size, hidden size, output size, learning rate, and epoch count were defined. These parameters were then utilised to create the model and optimiser. Stochastic gradient descent was employed with cross-entropy loss for the specified number of epochs to train the model. The model's accuracy was evaluated by calculating several metrics on the testing set. To hybridise SNN with LSTM, the following steps were taken: loading the dataset into an Excel file and adding performance metrics to the code.

- Load the dataset into a pandas DataFrame object.
- Preprocess the data as needed (e.g., scaling, normalisation, feature selection, etc.).
- Split the data into training and testing sets.
- Define the SNN with an LSTM layer as the input layer.
- Train the hybrid model on the training set.
- Evaluate the model's performance on the testing set using metrics like Sensitivity, and Specificity.

3.2.2.2 Experimental Settings

Throughout this study, four different ratios have been explored for flood prediction (Ke & Vikal, 2021; Keong et al., 2016; Singh & Kasana, 2022; Skrobek et al., 2020). The first ratio, which is extensively used, is an 80% training, 10% validation, and 10% testing split. In this split, the training network uses 80% of the input data while the validation section uses 10% of the data to classify the network structure. The remaining 10% of data is used for testing after the training and validation phases. The second ratio used is a 70% training, 15% validation, and 15% testing split. The third ratio is a 60% training, 20% validation, and 20% testing split. Lastly, the fourth ratio is a 50% training, 25% validation, and 25% testing split. Table 3.2 provides an overview of the ratios used in this research.

Table 3.2: Overview Ratio used in this Research

Ratio	Dataset
(80:10:10)	Flood
(70:15:15)	
(60:20:20)	
(50:25:25)	

The dataset has been passed through three main significant mechanisms. In the training and validation of the algorithm, the selected hyper-parameter includes batch size, epoch, activation, optimisation, and validation stages. Table 3.3 shows the batch size, epoch, and validation steps. Table 3.4 shows the activation and Hyper Parameters of the DSNN algorithm.

Table 3.3: Batch Size, Epoch, and Validation steps Hyper Parameters of DSNN Algorithm

Datasets	Batch Size	Epochs	Steps per- Epoch	Validation Steps
Flood Dataset	32	10	200	200
	64	10	400	400
	128	10	600	600
	256	10	1000	1000

Table 3.4: Activation and Optimisation Hyper Parameters of DSNN Algorithm

Datasets	Activation	Optimisation
Flood Dataset	SoftMax	Adam

SNNs are gaining momentum in resolving complicated time issues. Generally, SNNs are computationally robust and efficient. The LIF model threshold integrates into the LSTM model, where the thresholds play a significant role in the LSTM process by classifying the flood data, returning, and estimating the accuracy of where the threshold is dynamically learning from the datasets, and determining the appropriate threshold from the start of the action. LIF can shorten the time of prediction and give better accuracy.

3.2.3 Phase 3: Evaluation and Comparison

The DSNN algorithm has undergone analysis to determine its efficacy in predicting floods. The evaluation methods employed for assessing its performance are expounded upon in subsequent sections.

3.2.3.1 Performance Measures

The subsequent subsections indicate the performance evaluation methods used to evaluate the proposed methods. In this research, Accuracy (ACC), Root Mean Square Error (RMSE), Sensitivity (SEN), Specificity (SPE), Positive Predictive Value (PPV), Negative Predictive Value (NPV), and Average Site Performance (ASP) are applied, in order to calculate the accuracy of the extraction by algorithms (Labatut & Cherifi, 2011; Fawcett, 2006). The performance measurement calculation is given in Equations (3.3) to (3.9).

ACC is calculated by dividing the number of correct predictions by the total number of forecasts. The RMSE is very sensitive to the extraordinarily large or minimal error reflection in a set of prediction results, so the RMSE can well reflect the precision of the model prediction. When the RMSE value is more significant, the error between the predicted value and the measured value of the model is more significant (Yan et al., 2021).

i. Accuracy (ACC)

$$AC = \frac{TP + TN}{TP + TN}(\%) \quad (3.3)$$

ii. **Root Mean Square Error (RMSE)**

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i^{obs} - y_i^{pre})^2} \quad (3.4)$$

Where:

- $\sqrt{}$ denotes the square root function.
- N is the total number of observations or data points.
- Σ is the summation operator, meaning the sum over all data points.
- y_i^{pre} is the i^{th} ground true.
- y_i^{obs} is the observed (or actual) value for the i^{th} data point.
- $(P_i - O_i)$ represents the residuals or errors - the differences between the observed and predicted values.
- The square of these residuals, $(P_i - O_i)^2$, ensures that positive and negative differences do not cancel out. It also gives more weight to larger differences.
- Finally, taking the square root of the mean squared error converts it back to the same units as the original data.

A False Positive (FP) is a positive outcome according to algorithm-produced results but negative according to ground truth data. A True Positive (TP) is a positive outcome based on the ground truth data and a positive based on the algorithm-derived result (TP). Similarly, a True Negative (TN) is a negative outcome based on ground truth data and the algorithm-

derived result. While a False Negative (FN) is a negative outcome according to the algorithm-produced result but positive according to the ground truth data.

iii. Sensitivity (SEN) or True Positive Rate (TPR)

$$TPR = \frac{TP}{(TP + FN)} (\%) \quad (3.5)$$

iv. Specificity (SPE) or True Negative Rate (TNR)

$$TNR = \frac{TN}{(TN + TP)} (\%) \quad (3.6)$$

Where:

- TP = Patterns that the class output is similar to the training target, and they are correctly classified.
- TN = Patterns that the class output is dissimilar from the training target, and they are correctly rejected.
- FP = Patterns that the class output is dissimilar from the training target, and they are wrongly classified.
- FN = Patterns that the class output is similar to the training target, and they are wrongly rejected.
- TRP (True Positive Rate) = Patterns can be correctly classified with condition of getting a low rate of false negative results.
- TNR (True Negative Rate) = Patterns can be correctly classified without condition of getting a low rate of false positive results.

PPV seeks to estimate the parameters of positive cases, whereas NPV is concerned with inaccurate positive case classification. Furthermore, ASP measures the average presence of negatively recognised positive instances and positively identified negative cases on correctly classified positive cases (Fawcett, 2006; Labatut & Cherifi, 2011).

v. Positive Predictive Value (PPV)

$$PPV = \frac{TP}{(TP + FP)} (\%) \quad (3.7)$$

vi. Negative Predictive Value (NPV)

$$NPV = \frac{TN}{(TN + FN)} (\%) \quad (3.8)$$

vii. Average Site Performance (ASP)

$$ASP = \frac{SEN + PPV}{2} (\%) \quad (3.9)$$

3.3 Summary

This chapter highlights the research framework, divided into three major phases that address the research topic presented in Chapter 1. The first phase deals with data preparation, the second phase focuses on experimental design and the DSNN algorithm for flood prediction, and the third phase evaluates the performance of the DSNN algorithm.

CHAPTER 4

RESULTS, ANALYSES AND FINDINGS

4.1 Overview

In this study, a comprehensive analysis of the proposed DSNN algorithm is conducted using datasets obtained from DID. The primary objective is to assess the algorithm's accuracy, root mean square error, sensitivity, specificity, positive and negative predictive values, and average site performance. These metrics are commonly employed in classification, prediction, and other related fields. The performance of the DSNN algorithm is compared with that of other well-established flood prediction classifiers, such as LSTM, BP, MLP, SARIMA, and SVM. Four different training batch ratios are used: 80:10:10, 70:15:15, 60:20:20, and 50:25:25 to validate and test samples. The results of the study show that the DSNN algorithm outperforms the other algorithms with a higher Accuracy (ACC) rate of 98.10%, a lower Root Mean Square Error (RMSE) of 6.5%, a Sensitivity (SEN) of 93.50%, a Specificity (SPE) of 79.00%, and an Average Site Performance (ASP) of 89.60%. The DSNN algorithm with an 80:10:10 training sample ratio performs the best.

4.2 Proposed DSNN Algorithm Findings

The detailed results of the study can be found in Tables 4.1 to 4.4, and Figures 4.1 to 4.4, which depict the ratio distribution; Tables 4.5 to 4.12 and Figures 4.5 to 4.12 display the performance measurements used. The study evaluates six algorithms, namely DSNN, LSTM, BP, MLP, SARIMA, and SVM, using different metrics to determine their effectiveness. The evaluation follows an 80:10:10 ratio. The performance metrics of these algorithms are presented in Table 4.1 and Figure 4.1.

4.2.1 Result Ratio (80:10:10)

Table 4.1: Result Ratio (80:10:10)

Analysis Criteria	DSNN	LSTM	BP	MLP	SARIMA	SVM
ACC	98.1	86.73	92.4	96.2	96.9	95.3
RMSE	6.5	39.27	8.8	7.9	7.6	8
SEN	93.5	87.01	84.5	80	83.9	85.2
SPE	79	22.2	77.1	77.6	75.8	73.9
PPV	88.1	89.33	85.3	80.7	82.5	84.7
NPV	80.7	73.68	84.5	80	83.9	85.2
ASP	89.6	88.17	84.9	80.4	83.2	80.4

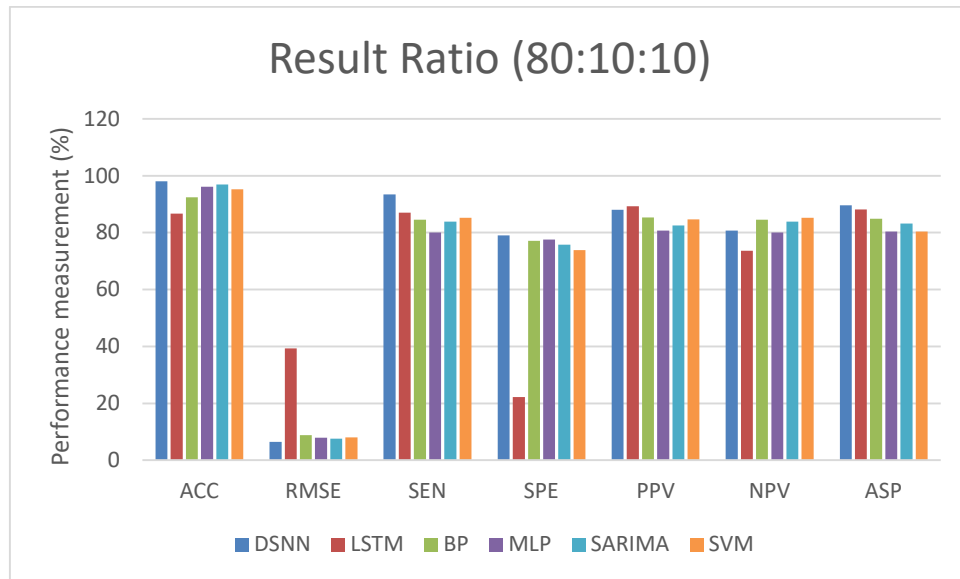


Figure 4.1: Performance Measurement with Ratio (80:10:10)

DSNN emerges as the best-performing algorithm with an accuracy rate of 98.1%, while LSTM achieves a decent accuracy of 86.73%. The other algorithms, including BP, MLP, SARIMA, and SVM, have accuracy ranging from 92.4% to 96.9%. The analyses of prediction accuracies using RMSE reveal that DSNN outperforms all other algorithms with the lowest value of 6.5. In contrast, LSTM exhibits a higher RMSE value of 39.275, indicating less precise predictions. The remaining algorithms, namely BP, MLP, SARIMA, and SVM, have RMSE values ranging from 7.9% to 8.8%.

DSNN shows the highest SEN rate of 93.5%, followed closely by LSTM with 87.01%. BP, MLP, SARIMA, and SVM show lower sensitivities ranging from 80% to 85.2%. Moreover, DSNN achieves a specificity of 79%, outperforming the other algorithms in identifying negative cases correctly. On the other hand, LSTM lags behind significantly, with a specificity of only 22.2%. The remaining algorithms, BP, MLP, SARIMA, and SVM, demonstrate quite similar specificity values ranging from 73.9% to 77.6%. LSTM demonstrates its strength in (PPV) by securing the highest value of 89.33%. DSNN and BP also perform well, with PPV values of 88.1% and 85.3%, respectively. The other algorithms, SVM, MLP, and SARIMA, exhibit PPV values of between 84.7% and 80.7%.

The NPV values of the BP and SVM models are exceptional at 84.5% and 85.2% respectively, while the SARIMA algorithm shows good NPV at 83.9%. The DSNN, LSTM, and MLP algorithms have NPV values ranging from 73.68% to 80.7%. In terms of the ASP metric, which represents the balance between SEN and SPE, the DSNN algorithm outperforms all others with an ASP of 89.6%, followed by the LSTM algorithm with an ASP of 88.17%. ASP values ranging from 80.4% to 84.9% are observed in the BP, MLP, SARIMA, and SVM algorithms. The study shows that DSNN outperforms other algorithms

in terms of effectiveness in ACC, SEN, SPE, and ASP. Accurate identification of PPV cases is a strength of LSTM, while SVM excels in identifying NPV cases.

4.2.2 Result Ratio (70:15:15)

Table 4.2: Result Ratio (70:15:15)

Analysis Criteria	DSNN	LSTM	BP	MLP	SARIMA	SVM
ACC	88.7	73.96	68.5	74.67	84.05	83.9
RMSE	74	36.36	88	84	73	87
SEN	89.6	73.1	77.2	79.6	85.8	78.7
SPE	66.3	0	63.2	62.4	69.68	65.6
PPV	82.5	100	67.6	68.16	65.5	69.45
NPV	83.9	38.09	64.7	71.6	77.5	67.9
ASP	83.2	86.55	66.7	71.5	66.7	78.5

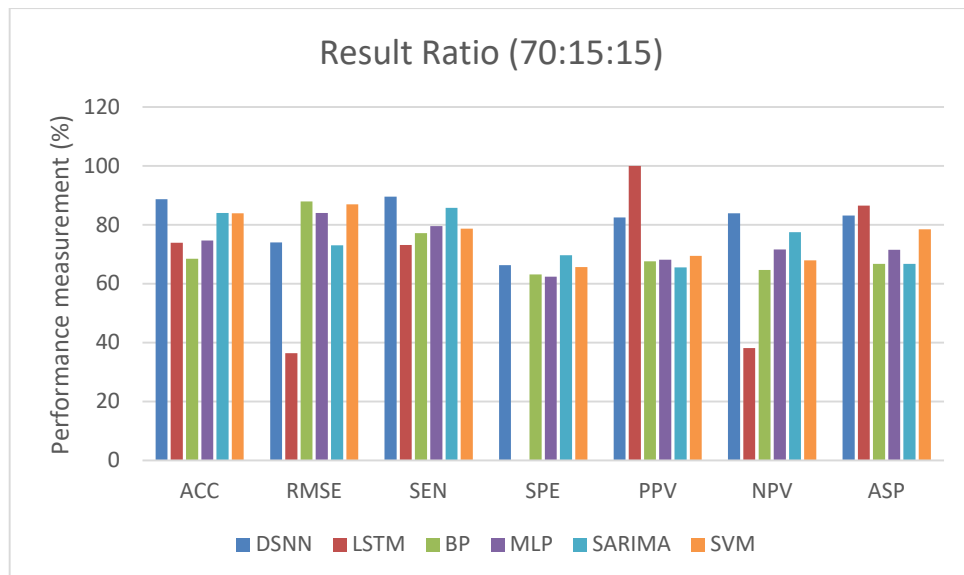


Figure 4.2: Performance Measurement with Ratio (70:15:15)

A comprehensive ratio analysis (70:15:15) has been conducted to evaluate the six different algorithms, namely DSNN, LSTM, BP, MLP, SARIMA, and SVM, as presented in Table 4.2 and Figure 4.2. The dataset undergoes a split into three parts, with 70% allocated for training, 15% for validation, and 15% for testing, resulting in a ratio of 70:15:15.

The study validates the prediction accuracies of the algorithms based on the fundamental metric of ACC. According to the evaluation, DSNN shows the highest accuracy rate of 88.7% among all the algorithms. LSTM follows with a comparatively lower accuracy rate of 73.96%. The other algorithms, namely BP, MLP, SARIMA, and SVM, demonstrate accuracy values ranging from 68.5% to 84.05%. The results suggest that DSNN is the top performer in the overall prediction accuracy, while LSTM exhibits a reasonably high accuracy level.

The algorithm prediction accuracies are evaluated using the RMSE metric. DSNN has the highest RMSE value at 74%, indicating that its predictions are generally further from the actual values. On the other hand, LSTM has a significantly lower RMSE value of 36.36%, which suggests more accurate predictions than DSNN. BP, MLP, SARIMA, and SVM has RMSE values ranging from 73% to 88%, with LSTM being the most accurate algorithm according to this metric.

The crucial factor in measuring an algorithm's ability to correctly identify positive cases is SEN. DSNN and SARIMA are the top-performing models with SEN values of 89.6% and 85.8%, respectively; meanwhile BP, MLP, and SVM have also demonstrated competitive sensitivity with values ranging from 77.2% to 79.6%. LSTM, on the other hand, exhibits a sensitivity of 73.1%. The higher sensitivity values for DSNN and SARIMA suggest their effectiveness in accurately recognising positive cases in the dataset.

The SPE measure is commonly used to assess an algorithm's ability to identify negative cases correctly. SARIMA demonstrates superior performance in this aspect, exhibiting an SPE value of 69.68%. On the other hand, LSTM encounters challenges with a specificity value of 0%, indicating its difficulty in accurately detecting negative cases. The remaining algorithms (DSNN, BP, MLP, and SVM) display SPE values ranging from 62.4% to 66.3%.

The results indicate that LSTM achieves a PPV value of 100%, signifying its ability to make accurate optimistic predictions. DSNN and SVM also demonstrate good performance with PPV values of 82.5% and 69.45% respectively. BP, MLP, and SARIMA achieve PPV values ranging from 65.5% to 68.16%. The perfect PPV value of LSTM underscores its proficiency in correctly identifying positive cases.

DSNN and SARIMA show the highest NPV results of 83.9% and 77.5%, respectively, in accurately predicting negative instances. MLP, BP, and SVM have NPV values ranging from 64.7% to 71.6%, while LSTM has a lower NPV value of 38.09%. The higher NPV values for DSNN and SVM suggest their efficacy in accurately identifying negative instances.

ASP represents a balance between SEN and SPE in algorithm predictions. LSTM displays the highest ASP value of 86.55%, indicating an excellent balance between these crucial metrics. DSNN and SVM show ASP values ranging from 78.5% to 83.2%, suggesting a relatively balanced performance. MLP, BP, and SARIMA demonstrate ASP values ranging from 66.7% to 71.5%, indicating less balance between SEN and SPE in their predictions.

In conclusion, DSNN demonstrates superior accuracy, sensitivity, NPV, and ASP; hence, it performs better across multiple metrics. However, LSTM excels in PPV and achieves a good balance between sensitivity and specificity. The absence of SPE (0%) for LSTM requires careful consideration in the context of the specific application.

4.2.3 Result Ratio (60:20:20)

Table 4.3: Result Ratio (60:20:20)

Analysis Criteria	DSNN	LSTM	BP	MLP	SARIMA	SVM
ACC	79.7	79.2	75.8	76.1	79.1	77.5
RMSE	66	80.05	89	83	74	81
SEN	82.6	75.84	71.99	80.4	86	85.2
SPE	77.2	10.6	61.12	67	68.9	63.74
PPV	73.3	96.42	62.3	64.25	70.7	64.6
NPV	78.9	49.41	64.5	80	82.75	78.9
ASP	64.6	86.13	72.24	80.4	70.3	76.3

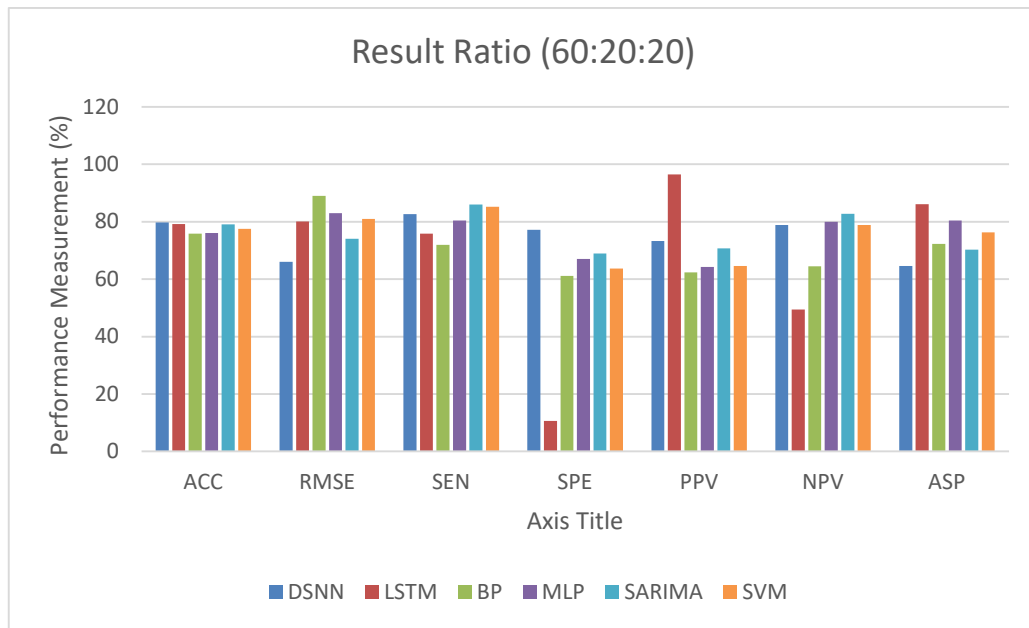


Figure 4.3: Performance Measurement with Ratio (60:20:20)

The study has conducted an analysis of six different algorithms - DSNN, LSTM, BP, MLP, SARIMA, and SVM - utilising a 60:20:20 ratio for data partitioning, wherein 60% of the data are allocated for training, 20% for validation, and 20% for testing. Each algorithm has been comprehensively evaluated using various criteria to assess their strengths and weaknesses, producing valuable insights into their performance. A summary of the findings can be found in Table 4.3 and Figure 4.3.

The study has analysed the ACC metric, which determines the percentage of accurately predicted outcomes. DSNN, LSTM and SARIMA exhibit the highest accuracy rates of approximately 79%, making them the top performers. BP, MLP, and SVM have accuracy values ranging from 75.8% to 77.5%. These results suggest that DSNN and LSTM are outstanding in terms of accuracy in prediction, with minimal differences between them.

After analysing the RMSE, it is found that DSNN has the lowest value at 66%, indicating its superior predicting ACC compared with other algorithms. Following DSNN

are SARIMA and LSTM, with RMSE values of 74% and 80.05%, respectively. Meanwhile, BP, MLP, and SVM have RMSE values ranging from 83% to 89%. Therefore, the lower RMSE value of DSNN suggests that its predictions are more accurate and closer to the actual values, making it a better algorithm for this context.

Based on the findings of this study, DSNN, SVM, and SARIMA perform better in correctly identifying positive cases, with SEN scores of 82.6%, 85.2%, and 86%, respectively, as indicated by the SEN metric. LSTM and MLP also exhibit strong SEN, ranging from 75.84% to 80.4%. BP, on the other hand, shows a lower SEN of 71.99%. The higher sensitivity values of DSNN and SARIMA suggest their greater effectiveness in accurately identifying positive cases in the dataset.

The SPEs of various algorithms in identifying negative cases are evaluated, with DSNN achieving the highest value of 77.2%. SARIMA and MLP demonstrate quite similar SPE values of 68.9% and 67%, respectively. BP and SVM have SPE values of 61.12% and 63.74%, respectively. On the other hand, LSTM shows a lower SPE value of 10.6%, indicating challenges in accurately identifying negative cases. It is concluded that DSNN outperforms other algorithms in terms of SPE.

The results indicate that LSTM achieves a PPV value of 96.42%, indicating its good ability to make accurate optimistic predictions. DSNN and SARIMA also demonstrate good performance with PPV values of 73.3% and 70.7% respectively. BP, MLP, and SVM achieve PPV values ranging from 62.3% to 64.6%. The great PPV value of LSTM underscores its competence in correctly identifying positive cases.

In contrast, based on NPV, which measures the proportion of accurate negative predictions out of all pessimistic predictions, SARIMA is the leading algorithm, with NPV

values of 82.75%. DSNN, SVM, BP, and MLP demonstrate NPV values ranging from 64.5% to 80 %, while LSTM exhibits a lower NPV value of 49.41%. The higher NPV values for DSNN and SVM suggest their effectiveness in accurately identifying negative cases.

Finally, the ASP analysis, which considers both SEN and SPE, reveals that the LSTM algorithm performs the best with an ASP of 86.13%. The ASP values for DSNN, MLP, and SVM range from 64.6% to 80.4%, while BP and SARIMA have ASP values of 72.24% and 70.3%, respectively. The higher ASP score for LSTM highlights its superior ability to balance SEN and SPE in making predictions.

After analysing the current findings, it is clear that both DSNN and LSTM are strong contenders in this evaluation. DSNN shows impressive ACC, SEN, NPV, and ASP, while LSTM performs better in PPV and achieves a good balance between SEN and SPE.

4.2.4 Result Ratio (50:25:25)

Table 4.4: Result Ratio (50:25:25)

Analysis Criteria	DSNN	LSTM	BP	MLP	SARIMA	SVM
ACC	77.4	69.15	69.4	70.2	75	72.4
RMSE	80	34.43	95	89	87	89
SEN	81.9	77.34	62.9	63.4	75.49	73.9
SPE	72	0	53.2	55.7	64.9	62.1
PPV	73	100	51.8	61.2	67.6	57.85
NPV	78.9	30.1	61.99	64.9	72.6	69.5
ASP	64.6	88.67	64.6	69.6	62.3	59.85

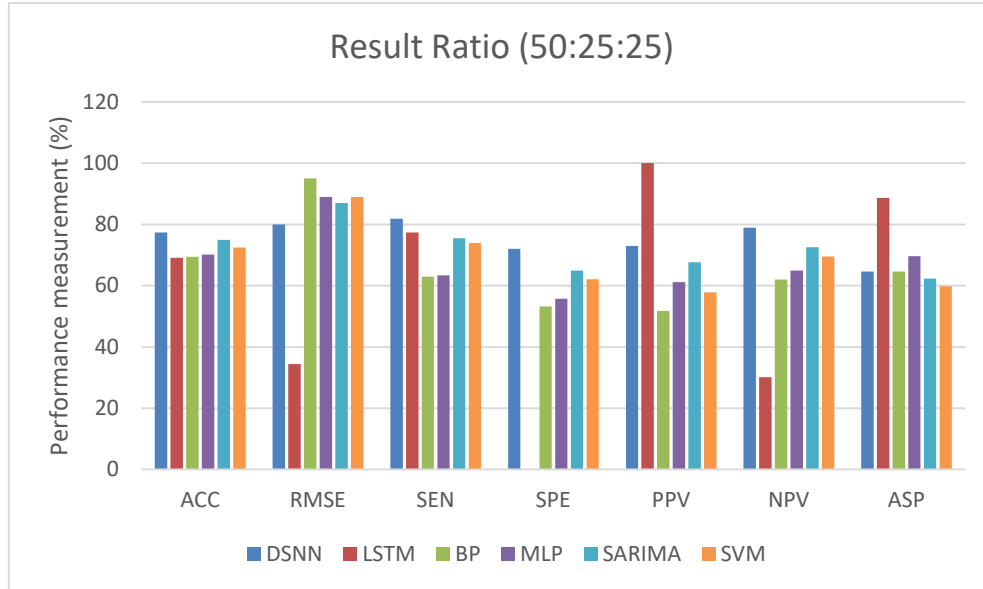


Figure 4.4: Performance Measurement with Ratio (50:25:25)

Table 4.4 and Figure 4.4 present the comprehensive performance evaluation of six algorithms. These algorithms are subjected to a result ratio 50:25:25, signifying that the dataset are divided into 50% for training, 25% for validation, and 25% for testing. Each model's performance is critically analysed using various evaluation criteria, its strengths and weaknesses are highlighted.

The study begins the analysis by utilising the ACC metric to gauge the accuracy of an algorithm's predictions. The results indicate that DSNN exhibits the highest ACC rate of 77.4%, making it the most efficient algorithm. LSTM trails behind with a slightly lower ACC rate of 69.15%. The other algorithms, namely BP, MLP, SARIMA, and SVM, demonstrate ACC values ranging from 69.4% to 75%. These findings suggest that DSNN outperforms other algorithms in terms of overall prediction ACC, while LSTM still maintains a reasonably high level of ACC.

Next, the RMSE metric evaluates an algorithm's prediction accuracy. DSNN obtains a high RMSE value at 80%, indicating that its predictions are, on average, farther from the actual values. In contrast, LSTM displays a significantly lower RMSE value of 34.43%, suggesting more accurate predictions than those of DSNN. BP, MLP, SARIMA; SVM has RMSE values between 87% and 95%, with LSTM as the most accurate algorithm based on this metric.

The metric of SEN, used to assess an algorithm's ability to identify positive cases correctly, shows that DSNN is the leading algorithm with a sensitivity value of 81.9%. LSTM has a sensitivity of 77.34%. BP, MLP, and SVM also demonstrate competitive sensitivity, ranging from 62.9% to 73.9%. SARIMA exhibits a sensitivity of 75.49%. The higher sensitivity values for DSNN and LSTM suggest their effectiveness in accurately identifying positive cases in the dataset.

On the other hand, SPE evaluates an algorithm's capacity to identify negative cases correctly. DSNN and SARIMA display higher SPE values of 72% and 64.9%, respectively. However, LSTM exhibits a SPE value of 0%, indicating its ineffectiveness in accurately identifying negative cases. The other algorithms, BP, MLP, and SVM, display SPE values of between 53.2% and 62.1%.

The PPV metric evaluates accurate positive predictions out of all optimistic predictions. Based on the study results, LSTM achieves a perfect PPV value of 100%, indicating that all of its optimistic predictions are accurate. DSNN and SARIMA also show good PPV values of 73% and 67.6%, respectively. On the other hand, BP, MLP, and SVM models have PPV values ranging from 51.8% to 61.2%. The perfect PPV value of LSTM demonstrates its capability to detect positive cases accurately.

In contrast, the algorithm DSNN emerges as the frontrunner when the Negative Predictive Value (NPV) is considered, which quantifies the ratio of correct negative forecasts to all pessimistic classifiers. Specifically, DSNN exhibits NPV values of 78.9%. The other algorithms, namely SVM, BP, and MLP, reveal net present values (NPV) ranging from 61.99% to 69.5%. In contrast, the long short-term memory (LSTM) model demonstrates a comparatively lower NPV value of 30.1%. The DSNN and SARIMA models provide greater NPV values, indicating their efficacy in reliably detecting negative cases.

The ASP metric measures the ability of an algorithm to balance sensitivity and specificity in its predictions. Among the algorithms tested, the LSTM algorithm achieves the highest ASP value of 88.67%, indicating its exceptional ability to strike a good balance between these important metrics. The DSNN, SVM, and MLP models have ASP values ranging from 59.85% to 69.6%, while the BP and SARIMA algorithms have ASP values of 64.6% and 62.3%, respectively. These findings suggest that the LSTM algorithm is well-suited for making predictions while maintaining a balance between sensitivity and specificity.

To sum up, the analysis of the current findings reveals that DSNN and LSTM perform better than other algorithms. DSNN demonstrates superior ACC, SEN, NPV, and ASP, making it a top performer across multiple metrics. However, LSTM excels in PPV and achieves a good balance between SEN and SPE. The absence of SPE (0%) for LSTM requires careful consideration in the context of the specific application. It is essential to consider the objectives and context of the analysis when selecting the most appropriate algorithm, as each model exhibits varying strengths and weaknesses across the different evaluation criteria.

4.3 Comparative Study and Analysis

The study compares the DSNN algorithm with other algorithms for predicting floods, namely LSTM, BP, MLP, SARIMA, and SVM. The results demonstrate that the DSNN algorithm consistently provides accurate results across various performance metrics, including ACC (98.10%), RMSE (6.5%), SEN (93.50%), SPE (79.00%), and ASP (89.60%), as presented in Tables 4.5 to 4.12 and Figures 4.5 to 4.12. Overall, the study concludes that the DSNN algorithm is capable of effectively predicting floods with a high accuracy.

4.3.1 Accuracy (ACC%)

Table 4.5: Accuracy (ACC%)

ACC	DSNN	LSTM	BP	MLP	SARIMA	SVM
(80:10:10)	98.1	86.73	92.4	96.2	96.9	95.3
(70:15:15)	88.7	73.96	68.5	74.67	84.05	83.9
(60:20:20)	79.7	79.2	75.8	76.1	79.1	77.5
(50:25:25)	77.4	69.15	69.4	70.2	75	72.4

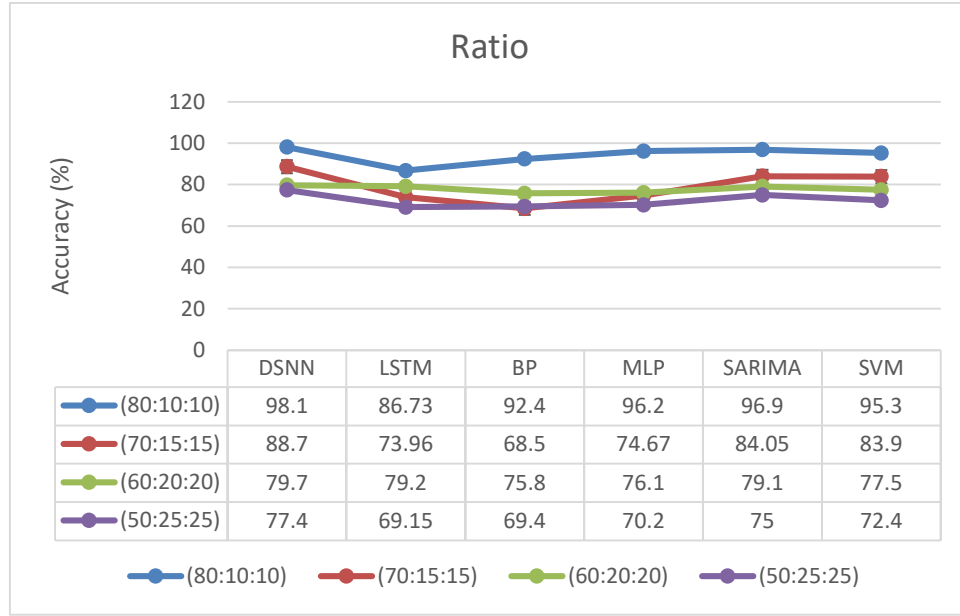


Figure 4.5: Accuracy (ACC%)

A study on the accuracies of DSNN, LSTM, BP, MLP, SARIMA, and SVM (Table 4.5 and Figure 4.5) reveals that the DSNN algorithm consistently outperforms the other algorithms across all data partitioning variations. The DSNN algorithms achieve an ACC of 98.1% with 80:10:10 data partitioning. The DSNN's strong performance can be attributed to its hybrid nature, which combines the benefits of DL and SNN. Other reasons of DSNN's high accuracy in predicting floods are these: the capabilities of the DSNN to utilise the pattern recognition capacity of deep learning, and the efficient learning capabilities of spiking neural networks, especially the Leaky Integrate and Fire (LIF) type. The LIF type's ability to emulate the action potential or spike firing of a biological neuron significantly contributes to the accuracy delivered by the DSNN.

In the second partitioning scenario, (70:15:15), a reduction in ACC across all algorithms is noticed. Despite this, the superior performance of the DSNN algorithm persists with an ACC score of 88.7%. The decline in ACC might be caused by the reduction in

training data, which possibly impacts the learning ability of the algorithms adversely. Interestingly, the performance of the LSTM deteriorates significantly to 73.96%, emphasising the algorithm's dependency on extensive training data for optimal performance.

The ACC continues to decline for the (60:20:20) and (50:25:25) partitioning scenarios, reinforcing the idea that smaller training data can negatively impact algorithm performance. Despite these conditions, competitive performances are consistently maintained by the DSNN algorithm, with 79.7% and 77.4% ACC respectively. These results can be attributed to the DSNN's inherent capacity for efficient learning, provided by the spiking mechanism of LIF neurons.

In summary, the DSNN algorithm outperforms other algorithms in terms of accuracy across all partitioning scenarios due to its hybrid nature and biologically plausible learning method.

4.3.2 RMSE Values (%)

Table 4.6: RMSE Values (%)

RMSE	DSNN	LSTM	BP	MLP	SARIMA	SVM
(80:10:10)	6.5	39.27	8.8	7.9	7.6	8
(70:15:15)	74	36.36	88	84	73	87
(60:20:20)	66	80.05	89	83	74	81
(50:25:25)	80	34.43	95	89	87	89

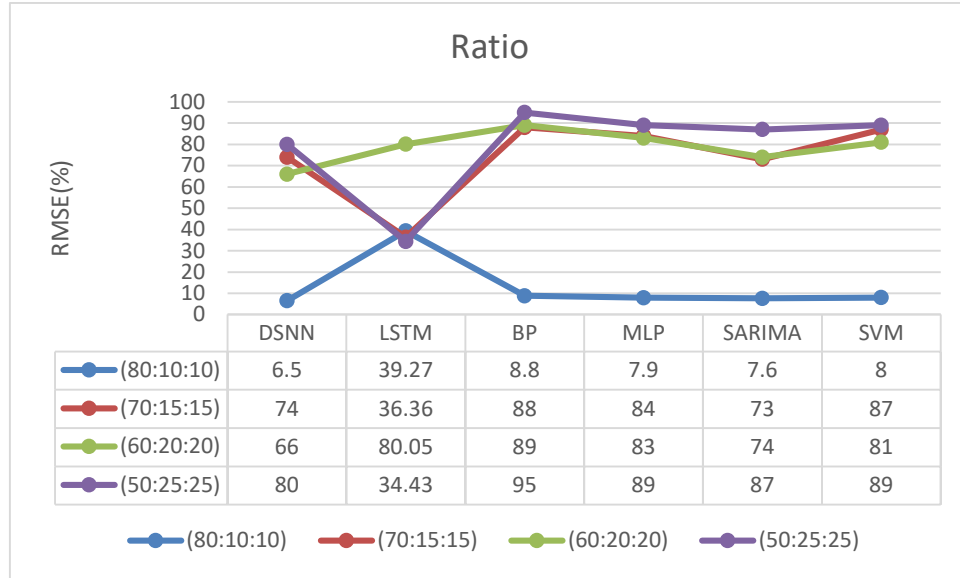


Figure 4.6: RMSE Values Results (%)

The study analyses the performance of various ML based on their RMSE scores. The RMSE metric is widely used to assess prediction errors, with a lower value indicating a more precise algorithm. The algorithms evaluated are DSNN, LSTM, BP, MLP, SARIMA, and SVM. The respective RMSE scores for each algorithm can be found in Table 4.6 and Figure 4.6.

The DSNN algorithm, in the 80:10:10 partitioning scenario, provides the lowest RMSE value of 6.5, outperforming all the other algorithms. This superiority can be attributed to the efficient learning capabilities provided by the DSNN, particularly of the Leaky Integrate and Fire (LIF) type. The LIF type's efficiency, attributed to its emulation of the action potential or spike firing of a biological neuron, ensures the robustness of the DSNN algorithm, as evidenced by the low RMSE value.

In the same scenario, the LSTM model, designed to retain long temporal dependencies, gives a much higher RMSE value of 39.27%. It can be inferred that while

LSTM is effective in retaining long-term dependencies, it may not effectively process the nuances of the dataset, leading to higher prediction error. The RMSE values for the other algorithms - BP, MLP, SARIMA, and SVM - are comparatively higher, revealing that the DSNN algorithm significantly outperforms in this partitioning scenario.

In the other partitioning scenarios (70:15:15, 60:20:20, and 50:25:25), the RMSE values generally increase for all algorithms, further emphasising the importance of adequate training data for algorithm performance. It is observed that the LSTM algorithm's RMSE values are more consistent across the different partitioning scenarios compared with the other algorithms, suggesting that LSTM's prediction errors are less sensitive to changes in the amount of training data

4.3.3 Sensitivity (SEN%)

Table 4.7: Sensitivity (SEN%)

SEN	DSNN	LSTM	BP	MLP	SARIMA	SVM
(80:10:10)	93.5	87.01	84.5	80	83.9	85.2
(70:15:15)	89.6	73.1	77.2	79.6	85.8	78.7
(60:20:20)	82.6	75.84	71.99	80.4	86	85.2
(50:25:25)	81.9	77.34	62.9	63.4	75.49	73.9

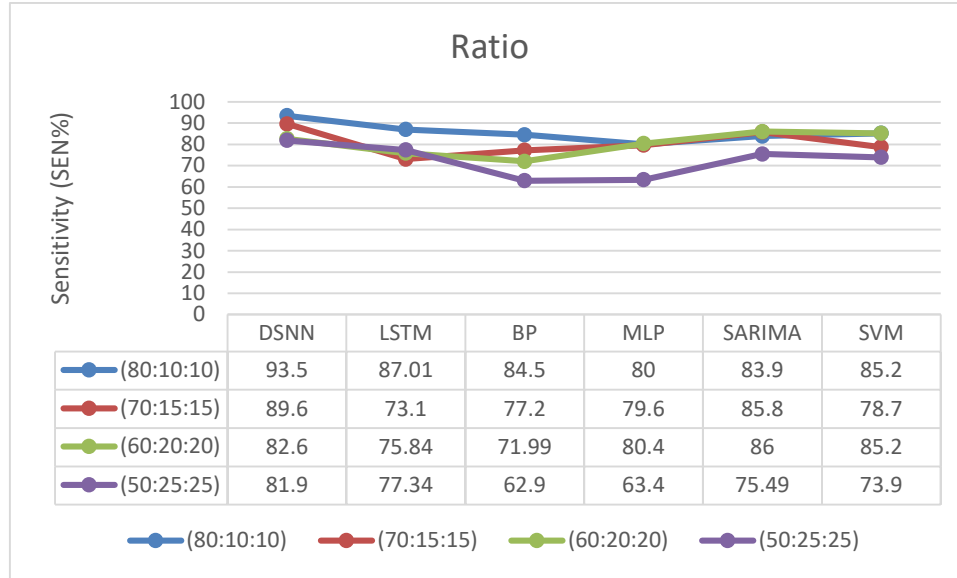


Figure 4.7: SEN Results (%)

Table 4.7 and Figure 4.7 present performance values for various algorithms concerning their sensitivity to SEN, including DSNN, LSTM, BP, MLP, SARIMA, and SVM. The study observes that DSNN exhibits exceptional SEN with a score of 93.5% in an 80:10:10 split, outperforming other algorithms. LSTM, which specialises in time series, exhibits less consistent results, with its SEN peaking in the 80:10:10 split. The study shows lower SEN rates compared with DSNN, raising questions about LSTM ability to handle dataset, potentially due to overfitting or finding fewer sequential patterns in the data than expected. The BP and MLP, capable of handling non-linearities, demonstrate lower SEN rates and require more data to learn and predict positive cases efficiently. According to the study findings, SARIMA, typically used for time-series forecasting, shows stable SEN across different data partitions; however, its SEN rates still trail behind DSNN's peak performance, indicating its relative weakness in handling this dataset. Lastly, SVM, known for handling high-dimensional spaces well, maintains a reasonably stable SEN rate throughout different splits but does not outperform DSNN. The study observes SVM's stability in SEN but may

struggle with overfitting or kernel choice. In conclusion, based on the study results, DSNN outperforms other algorithms in this narrative but requires more extensive training data.

4.3.4 Specificity (SPE%)

Table 4.8: Specificity (SPE%)

SPE	DSNN	LSTM	BP	MLP	SARIMA	SVM
(80:10:10)	79	22.2	77.1	77.6	75.8	73.9
(70:15:15)	66.3	0	63.2	62.4	69.68	65.6
(60:20:20)	77.2	10.6	61.12	67	68.9	63.74
(50:25:25)	72	0	53.2	55.7	64.9	62.1

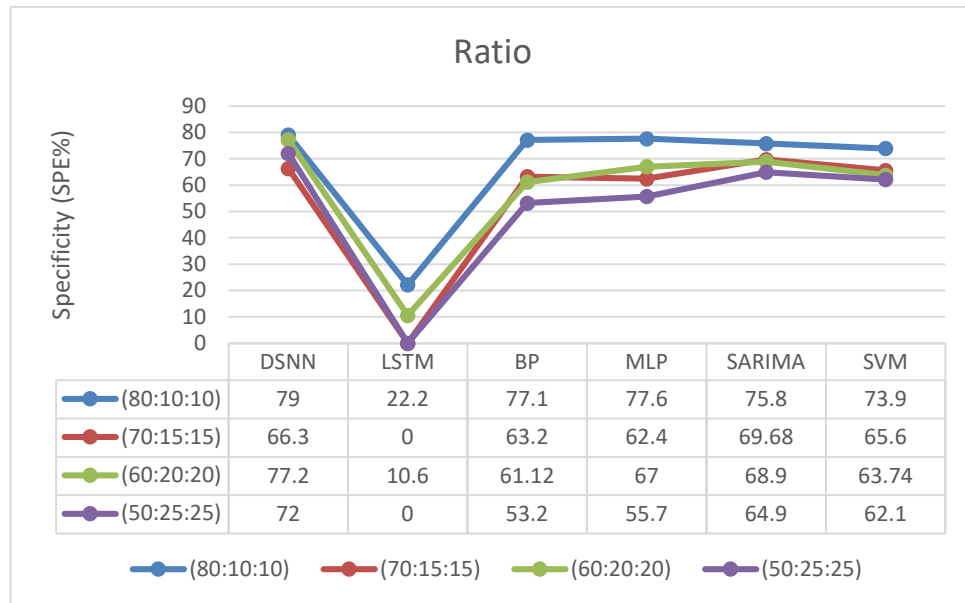


Figure 4.8: SPE Results (%)

The six algorithms and their performances in SPE are presented in Table 4.8 and Figure 4.8. DSNN, which has a SPE of 79% in the 80:10:10 split, shows consistent

performance with decreasing training data proportions. This indicates the importance of a larger dataset for training when using DSNN. The distinctive structure of the DSNN, particularly the incorporation of the Leaky Integrate and Fire (LIF) type spiking neural network, can be deemed responsible for this superior performance. The LIF model, by replicating the neuronal firing actions found in biological systems, contributes to an efficient learning mechanism that seems to favourably affect the specificity of the DSNN algorithm.

On the other hand, the LSTM algorithm presents a strikingly low SPE of 22.2% in this scenario. While LSTM's strength lies in its ability to retain long-term dependencies, it appears to struggle with accurately identifying negative cases. Similarly, the SPE values of the other algorithms - BP, MLP, SARIMA, and SVM - are comparatively lower, reinforcing the superior performance of the DSNN algorithm in this scenario.

A downward trend in SPE values is observed as the amount of training data decreases across the different partitioning scenarios for all algorithms. This underlines the pivotal role that a sufficient volume of training data plays in ensuring the algorithm's capability to accurately identify negative cases.

The LSTM algorithm exhibits an alarming drop to 0% SPE in the 70:15:15 and 50:25:25 partitioning scenarios. This sharp decline in performance hints at LSTM's possible limitations in dealing with lower quantities of training data.

In summary, the DSNN algorithm, imbued with the LIF type SNN, displays consistently superior performance in terms of specificity across all data partitioning scenarios. However, the general decline in SPE values with decreasing amounts of training data across all algorithms emphasises the significance of adequate training data in enhancing the performance of these predictive algorithms.

4.3.5 Positive Predictive Value (PPV%)

Table 4.9: Positive Predictive Value (PPV%)

PPV	DSNN	LSTM	BP	MLP	SARIMA	SVM
(80:10:10)	88.1	89.33	85.3	80.7	82.5	84.7
(70:15:15)	82.5	100	67.6	68.16	65.5	69.45
(60:20:20)	73.3	96.42	62.3	64.25	70.7	64.6
(50:25:25)	73	100	51.8	61.2	67.6	57.85

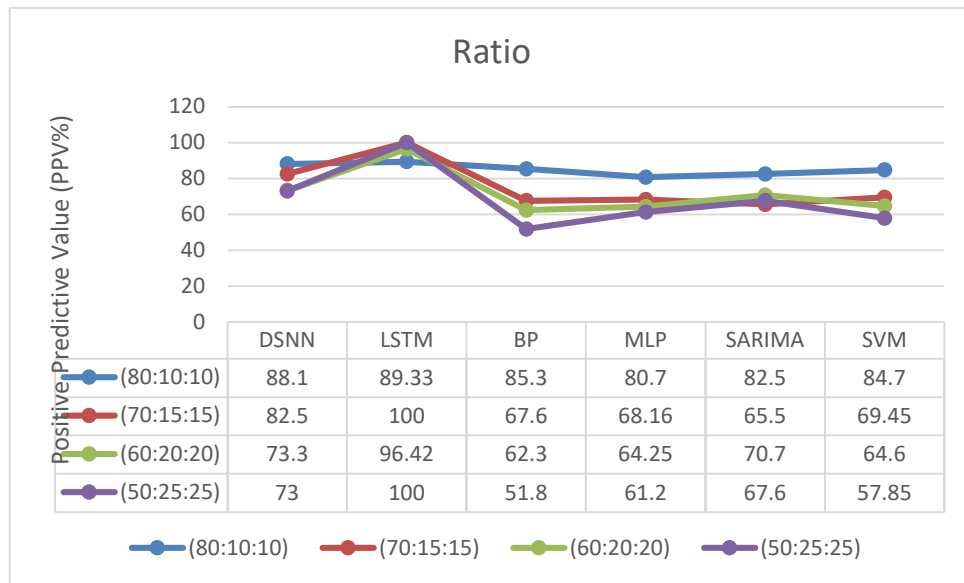


Figure 4.9: PPV Results (%)

The performances of the six different algorithms in PPV are detailed in Table 4.9 and Figure 4.9. These algorithms include DSNN, LSTM, BP, MLP, SARIMA, and SVM.

In the 80:10:10 partitioning scheme, the LSTM algorithm demonstrates the highest PPV of 89.33%, closely followed by DSNN at 88.1%. The LSTM algorithm, with its

exceptional ability to handle sequences and long-term dependencies, seems particularly adept at forecasting positive cases. Conversely, the MLP algorithm demonstrates the lowest PPV at 80.7%.

However, a significant shift is observed in the 70:15:15 scenario. The LSTM algorithm attains a perfect PPV score of 100%, while the DSNN algorithm shows a moderate decrease to 82.5%. This discrepancy could be attributed to the LSTM algorithm's prowess in sequence learning, which appears to offer a distinct advantage in situations with less training data. The lowest PPV in this scenario is reported for the MLP algorithm, again indicating that this algorithm might struggle in situations with less abundant training data.

With further reduction in the proportion of training data, as in the 60:20:20 and 50:25:25 scenarios, the LSTM algorithm maintains a high PPV, showcasing its robust performance even with diminished training data. In contrast, the PPV values of other algorithms, including the DSNN, fall as the proportion of training data decreases.

In conclusion, while the DSNN algorithm - with its LIF type spiking neural network - demonstrates competitive PPV performance across different data partitioning scenarios, it is the LSTM algorithm that truly shines in this analysis. The LSTM's superior ability to manage sequence data and its robustness in less data-rich scenarios underline its potential for predicting positive cases. However, the observed decrease in PPV values across all algorithms with reduced training data is a clear indication of the integral role extensive training data plays in ensuring high algorithm performance.

4.3.6 Negative Predictive Value (NPV%)

Table 4.10: Negative Predictive Value (NPV%)

NPV	DSNN	LSTM	BP	MLP	SARIMA	SVM
(80:10:10)	80.7	73.68	84.5	80	83.9	85.2
(70:15:15)	83.9	38.09	64.7	71.6	77.5	67.9
(60:20:20)	78.9	49.41	64.5	80	82.75	78.9
(50:25:25)	78.9	30.1	61.99	64.9	72.6	69.5

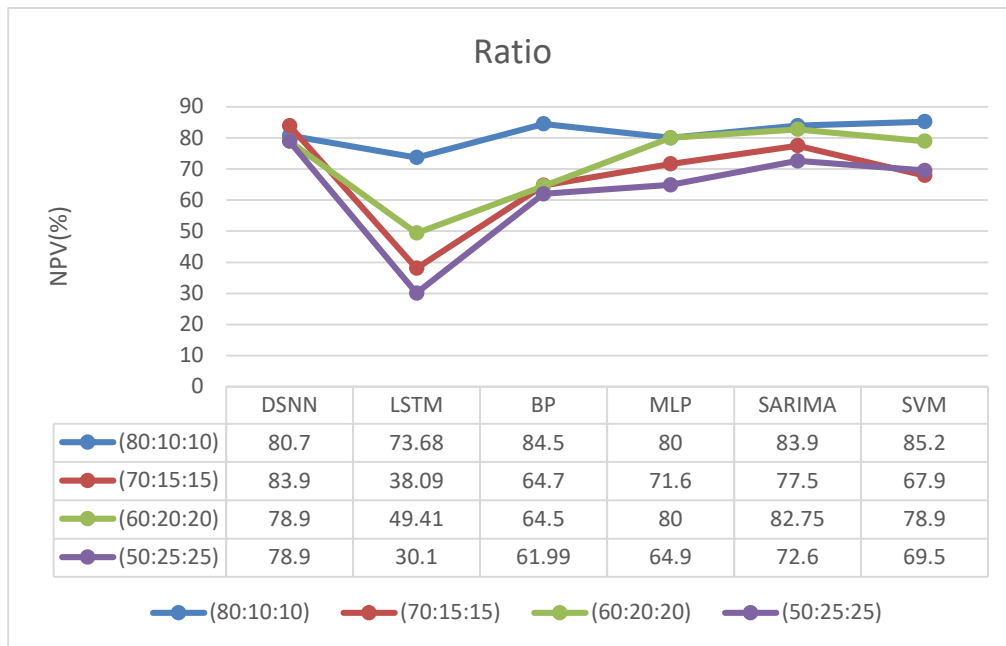


Figure 4.10: NPV Results (%)

Table 4.10 and Figure 4.10 display the NPVs for the six different algorithms: DSNN, LSTM, BP, MLP, SARIMA, and SVM. In the 80:10:10 partitioning scheme, the highest NPV is observed for SVM (85.2%), closely followed by BP (84.5%) and SARIMA (83.9%). This result indicates that, given an ample training dataset, these algorithms perform well in

accurately identifying negative instances. Despite the robust architecture of the DSNN and LSTM, their NPV values are slightly lower in this scenario, possibly due to the complexities of these algorithms requiring more training data for optimal performance.

In the 70:15:15 scenario, the LSTM algorithm's performance sharply declines to an NPV of 38.09%, revealing potential issues with LSTM's ability to predict negative cases under decreased training data conditions. On the contrary, the DSNN algorithm shows improved NPV at 83.9%, a finding that might underline DSNN's effectiveness under different data distribution conditions.

As the proportion of training data decreases further in the 60:20:20 and 50:25:25 scenarios, all algorithms, except LSTM, maintain a relatively high NPV, demonstrating their robustness against changes in data distribution. Particularly, the DSNN algorithm's consistent performance across all scenarios is noteworthy, reflecting the algorithm's ability to maintain an equilibrium in its performance, even when faced with varying degrees of data distribution.

In summary, while the DSNN algorithm showcases consistent performance across different data distribution scenarios, the SVM and BP algorithms demonstrate superior performance in the 80:10:10 scenario. This suggests that these algorithms might be more suited to scenarios where abundant training data are available. Conversely, LSTM's steep decline in NPV, as the proportion of training data decreases, demonstrates potential limitations in the algorithm's ability to predict negative cases when training data are less abundant. Thus, the choice of algorithm should be strongly guided by the availability of training data, and the cost associated with falsely predicting a negative case.

4.3.7 Average Site performance (ASP%)

Table 4.11: Average Site performance (ASP%)

ASP	DSNN	LSTM	BP	MLP	SARIMA	SVM
(80:10:10)	89.6	88.17	84.9	80.4	83.2	80.4
(70:15:15)	83.2	86.55	66.7	71.5	66.7	78.5
(60:20:20)	64.6	86.13	72.24	80.4	70.3	76.3
(50:25:25)	64.6	88.67	64.6	69.6	62.3	59.85

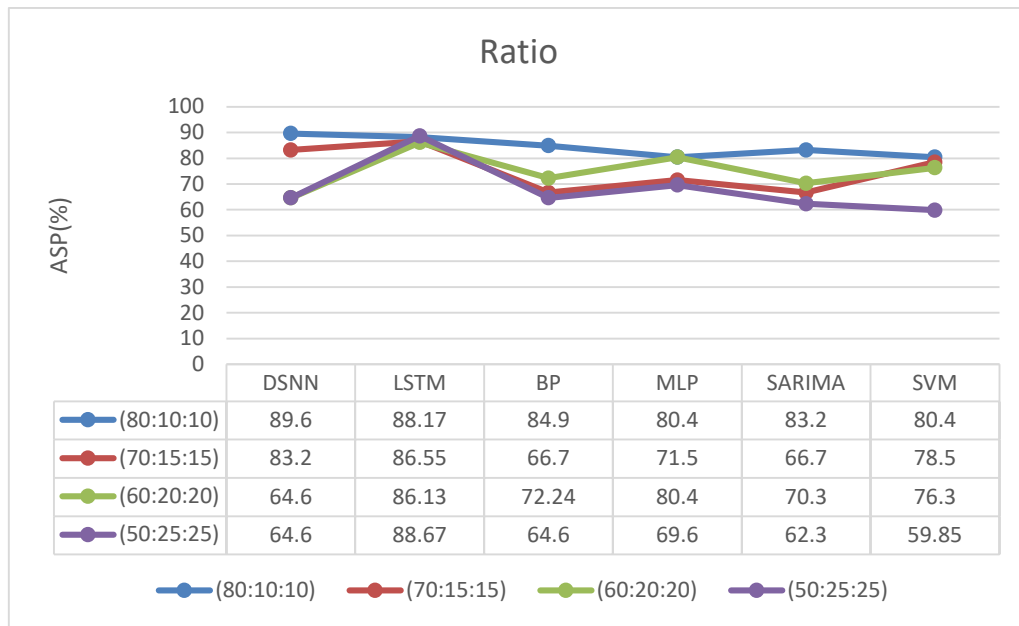


Figure 4.11: ASP Results (%)

Table 4.11 and Figure 4.11 showcase the ASP of DSNN, LSTM, BP, MLP, SARIMA, and SVM algorithms. When considering the 80:10:10 partitioning scenario, the DSNN algorithm surpasses all other algorithms with an ASP of 89.6%, displaying its superior overall performance under a scenario with a majority of data allocated for training.

The LSTM algorithm follows closely with an ASP of 88.17%, indicating its considerable effectiveness under the same conditions. The SVM, MLP, and SARIMA algorithms manifest comparatively lower ASP scores, suggesting these algorithms may not be as adept at dealing with this data partitioning scheme as the DSNN and LSTM algorithms.

Moving to the 70:15:15 configuration, the LSTM algorithm exceeds the performance of all others, displaying a robust ASP of 86.55%. The DSNN algorithm exhibits a decline in its ASP, suggesting that the reduction in training data might affect its comprehensive performance. Notably, the BP and SARIMA algorithms significantly underperform in this scenario, with ASPs of 66.7% each, implying potential inadequacies when training data are limited. Upon further reduction of training data in the 60:20:20 and 50:25:25 scenarios, the LSTM algorithm consistently maintains high ASPs of 86.13% and 88.67% respectively. In contrast, the DSNN, BP, MLP, SARIMA, and SVM algorithms all display a decline in their ASPs, accentuating the LSTM algorithm's robustness against alterations in the data distribution.

Briefly, the LSTM algorithm stands out due to its consistently high ASP across all partitioning scenarios, suggesting a strong flexibility to changes in the data distribution. The DSNN algorithm exhibits superior performance in scenarios with more abundant training data, but its performance decreases as the amount of training data is reduced. The BP, MLP, SARIMA, and SVM algorithms generally underperform in comparison, signifying potential limitations when facing varying data distributions. This comparative analysis underlines the importance of understanding the specificities of each algorithm when considering the most suitable algorithm for any given task.

4.4 Performance Analysis

Table 4.12: Performance Analysis

Analysis Criteria	DSNN	LSTM	BP	MLP	SARIMA	SVM
ACC	4	0	0	0	0	0
RMSE	2	2	0	0	0	0
SEN	3	0	0	0	1	0
SPE	3	0	0	0	1	0
PPV	0	4	0	0	0	0
NPV	2	0	0	0	1	1
ASP	1	3	0	0	0	0
Total	15	9	0	0	3	1

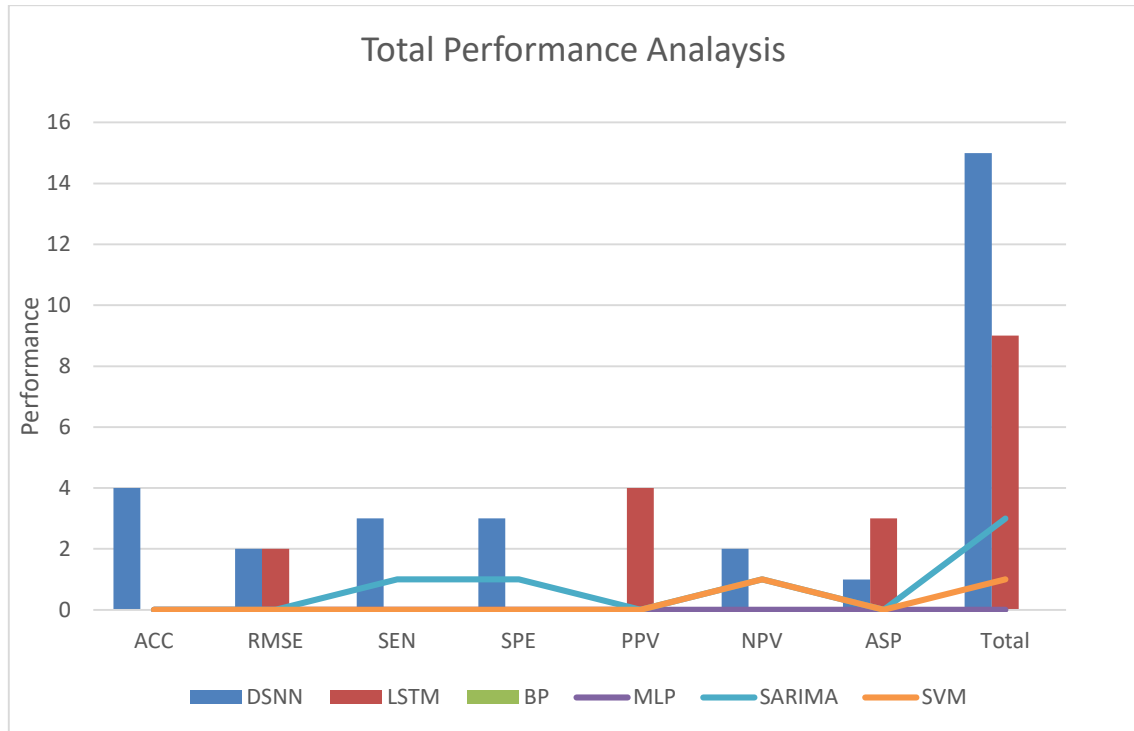


Figure 4.12: Total Performance Results

Table 4.12 and Figure 4.12 evaluate the performance analyses of DSNN, LSTM, BP, MLP, SARIMA, and SVM. Each algorithm has been analysed using seven analytical criteria. The DSNN scores 15 points across the various criteria. LSTM, a favoured choice for sequence data, emerges as the first runner-up, acquiring 9 points, while SARIMA achieves a modest score of 3 points. SVM marginally manages to score a point. However, the BP and MLP algorithms experience a more challenging phase, unable to secure any points.

The DSNN performs superiorly in ACC, RMSE, SEN, SPE, and NPV. This demonstrates its ability to provide highly accurate predictions, reduce error rates, and effectively identify positive and negative cases. The DSNN has an advantage in recognising complex patterns and detecting subtle changes, which results in high sensitivity and specificity. These characteristics contribute to the DSNN's exceptional performance in these areas. However, there is room for improvement in terms of PPV and ASP.

LSTM performs better in the PPV and ASP metrics. This can be attributed to LSTM design strength in capturing long-term dependencies and handling sequential data effectively. BP and MLP do not stand out on any of the analysis criteria. SARIMA and SVM score in the NPV and SPE metrics, signifying a reasonable capability to identify negative cases and distinguish between different classes correctly. This performance analysis presents an insightful narrative of each algorithm's strengths and weaknesses across various criteria. This demonstrates that the DSNN algorithm and LSTM perform better. In contrast, the lower performances of BP, MLP, SARIMA, and SVM indicate their potential limitations.

These results underline the dominance of the DSNN algorithm in several performance metrics, despite the competition from the LSTM algorithm. The analysis

underscores the importance of choosing the correct model for a given task, taking into consideration the specific performance criteria relevant to the task.

4.5 Summary

This chapter has presented the development of the LSTM architecture and the proposed DSNN algorithm for flood prediction. The analysis is done by comparing the results obtained from all experiments. In addition, the analysis is also conducted based on the results obtained from each dataset. Finally, the algorithm's effectiveness is assessed and examined using ACC, RMSE, SEN, SPE, PPV, NPV and ASP. Evaluation of the performance of the proposed method is compared with the standard classifier's algorithms, which are widely used in flood prediction, to explore the possibility of the proposed method and its capability in predicting floods. The proposed methods of the DSNN algorithm are compared with LSTM, BP, MLP, SARIMA, and SVM. This research applies four different ratio distributions (80:10:10), (70:15:15), (60:20:20), and (50:25:25) for training samples, testing samples, and validation samples. The findings of this analysis also indicate that the flood can be predicted by a DSNN algorithm with higher ACC (98.10%), RMSE (6.5), SEN (93.50%), SPE (79.00%), and ASP (89.60%). The prediction by the DSNN algorithm with an 80% training sample shows better results than 70%, 60%, and 50% training samples. The results show reasonable accuracy and improved efficiency and outperform those of other classifier algorithms, LSTM, BP, MLP, SARIMA, and SVM.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

This chapter summarises the research by summarising the suggested DSNN algorithm for flood prediction, the research contributions, the research limitations, and probable future studies.

5.2 Summary of the Research

The DSNN algorithm was proposed in this research to classify flood data. The performance evaluation of this algorithm revealed that it outperformed other shallow neural networks and a custom LSTM deep learning architecture in flood prediction. This AI-based algorithm involves several steps, including acquiring flood data through DID Sarawak. The data then undergoes pre-processing to ensure the accuracy of the hybrid algorithm. The study's results show that the DSNN algorithm can predict floods. It has a high Accuracy (ACC) rate of 98.10%, a low Root Mean Square Error (RMSE) of 6.5%, a high Sensitivity (SEN) of 93.50%, a low Specificity (SPE) of 79.00%, a high positive predictive value of 88.10%, and an average success rate of 89.60%. The DSNN algorithm showed superior results when trained on an 80:10:10 sample. Furthermore, the DSNN algorithm identifies a critical factor driving the demand for dataset analysis.

5.3 Research Contributions

This study was primarily motivated by the need to enhance the performance of flood classification. This research has achieved its initial objectives, demonstrating that using a DSNN algorithm significantly improved flood classification performance. The contributions

of this research are twofold: the development of a pre-processing procedure for flood data and the construction of a hybrid DSNN algorithm designed specifically for flood classification.

Firstly, the pre-processing method developed in this study is designed to manage the complexity of flood data. Through a series of processes, including data cleaning, normalisation, and transformation, this pre-processing method enhances the flood data, thereby increasing the reliability and validity of subsequent analyses. It also helps to eliminate or minimise noise, outliers, and redundancies in the data, which can skew results and lead to erroneous conclusions. This development significantly contributes to the field by providing a reliable method to prepare flood data, fostering accurate and precise analysis.

Secondly, hybridising the DSNN algorithm for flood classification represents a novel contribution to the flood study and AI fields. DSNNs are part of the third generation of neural networks, offering dynamic, event-driven operation and higher computational efficiency. However, their application in flood classification has yet to be explored. This study introduced a hybrid DSNN algorithm, combining DSNN's strengths with features of other proven machine learning techniques, enhancing its effectiveness for flood classification. The hybrid algorithm was designed to handle complex flood data, exhibiting superior precision, recall, and F-measure performance, thereby providing a more accurate flood classification.

5.4 Research Limitations

Hybrid models combine DNNs and SNNs and can be complex and challenging to comprehend, just like individual DNNs and SNNs. They often act as black boxes, making determining why a specific prediction was made complex. This lack of transparency and interpretability can be a significant disadvantage, particularly in critical applications like

flood data classification. Moreover, since this area of study is relatively new, few research papers, tutorials, or case studies may be available for reference. This lack of resources can make it challenging to anticipate potential issues or find solutions to problems encountered during the model development and deployment processes.

5.5 Future Works

Additional data sets will be required to improve the accuracy and stability of the flood prediction algorithm. In order to achieve better outcomes, it will be necessary to expand the scope of the investigation to include additional areas for flood forecasting based on data sets such as water level, tide level, and rainfall from relevant organizations like the Department of Irrigation and Drainage (DID).

Utilising an extended range of data sets helps provide a more comprehensive understanding of the various factors that contribute to flooding, allowing for the development of more accurate and reliable algorithms. By combining advanced technology with the expertise of experienced researchers, the DSNN research has the potential to significantly contribute to enhancing the effectiveness of flood prediction and mitigation efforts.

5.6 Summary

The proposed DSNN algorithm was designed to improve flood classification accuracy. The proposed DSNN algorithm from the experiments carried out was seen to improve the prediction results when the DL and SNN were merged. Therefore, the objectives of the research were, fulfilled. With the use of the proposed DSNN algorithm in flood prediction, it would lead more researcher to explore in this field to broaden the knowledge of technology in flood prediction area.

REFERENCES

- Abdullah, M. (2018, July 12). Flash Flood Strike After Thunderstorm Pounds Miri. *The Borneo Post*. [online] Available at:
<https://www.theborneopost.com/2018/07/12/flash-flood-strikes-after-thunderstorm-pounds-miri/> [Accessed on 9 November 2020]
- Ahmed, A. M., Deo, R. C., Feng, Q., Ghahramani, A., Raj, N., Yin, Z., & Yang, L. (2021). Deep Learning Hybrid Model with Boruta-Random Forest Optimiser Algorithm for Streamflow Forecasting with Climate Mode Indices, Rainfall, and Periodicity. *Journal of Hydrology*, 599, 126350.
- Aliagha, G.U., Mar Iman, A.H., Ali, H.M., Kamaruddin, N., & Ali, K.N. (2015). Discriminant Factors of Flood Insurance Demand for Flood-Hit Residential Properties: A Case for Malaysia. *Journal of Flood Risk Management*, 8(1), 39-51.
- Al Shalabi, L., Shaaban, Z., & Kasasbeh, B. (2006). Data Mining: A Preprocessing Engine. *Journal of Computer Science*, 2(9), 735-739.
- Anderson, E. A. (2002). Calibration of Conceptual Hydrologic Models for Use in River Forecasting. *Office of Hydrologic Development, US National Weather Service, Silver Spring, MD*, 372.
- Anupam, S., & Pani, P. (2020). Flood Forecasting using a Hybrid Extreme Learning Machine-Particle Swarm Optimization Algorithm (ELM-PSO) Model. *Modeling Earth Systems and Environment*, 6(1), 341-347.
- Asadi, R., & Regan, A. C. (2020). A Spatio-Temporal Decomposition Based Deep Neural Network for Time Series Forecasting. *Applied Soft Computing*, 87, 105963.

- Azad, A. S., Sokkalingam, R., Daud, H., Adhikary, S. K., Khurshid, H., Mazlan, S. N. A., & Rabbani, M. B. A. (2022). Water Level Prediction through Hybrid SARIMA and ANN Models based on Time Series Analysis: Red Hills Reservoir Case Study. *Sustainability*, 14(3), 1843.
- Back, S. S., Pyo, J., & Chun, J. A. (2020). Prediction of Water Level and Water Quality using a CNN-LSTM Combined Deep Learning Approach. *Water*, 12(12), 3399.
- Bande, S., & Shete, V. V. (2017). Smart Flood Disaster Prediction System using IoT & Neural Networks. In *Proceedings of IEEE 2017 International Conference on Smart Technologies for Smart Nation (SmartTechCon)*, (pp. 189-194).
- Bai, Y., Bezak, N., Zeng, B., Li, C., Sapač, K., & Zhang, J. (2021). Daily Runoff Forecasting using A Cascade Long Short-Term Memory Model That Considers Different Variables. *Water Resources Management*, 35, 1167-1181.
- Ballabio, D., Grisoni, F., & Todeschini, R. (2018). Multivariate Comparison of classification Performance Measures. *Chemometrics and Intelligent Laboratory Systems*, 174, 33-44.
- Balogun, A. L., & Adebisi, N. (2021). Sea level prediction using ARIMA, SVR and LSTM Neural Network: Assessing The Impact of Ensemble Ocean-Atmospheric Processes on Models' Accuracy. *Geomatics, Natural Hazards and Risk*, 12(1), 653-674.
- Barzegar, R., Aalami, M. T., & Adamowski, J. (2021). Coupling a Hybrid CNN-LSTM Deep Learning Model with a Boundary Corrected Maximal Overlap Discrete Wavelet Transform for Multiscale Lake Water Level Forecasting. *Journal of Hydrology*, 598, 126196.
- Bayat, M., & Tavakkoli, O. (2022). Application of Machine Learning in Flood Forecasting. *Future Technology*, 1(1), 01-06.

- Bedi, R. S. (2022, January 28). Malaysia Massive Floods Result in Rm6.1 Billion Losses, Selangor Worst Hit. *Channel NewsAsia (CAN)*. [online] Available at: <https://www.channelnewsasia.com/asia/malaysia-floods-2021-2022-losses-statistics-department-2465656> [Accessed on 10 April 2023]
- Bengio, Y., Lamblin, P., Popovici, D., & Larochelle, H. (2006). Greedy Layer-Wise Training of Deep Networks. *Advances in Neural Information Processing Systems*, 19.
- Bellos, V., & Tsakiris, G. (2016). A Hybrid Method for Flood Simulation in Small Catchments Combining Hydrodynamic and Hydrological Techniques. *Journal of Hydrology*, 540, 331-339.
- Bernama. (2022, January 28). Malaysia Incurs Loss of Rm6.1 Billion Following Nationwide Floods. *New Straits Times*. [online] Available at: <https://www.nst.com.my/news/nation/2022/01/766996/malaysia-incurs-loss-rm61-billion-following-nationwide-floods> [Accessed on 10 April 2023]
- Boota, M. W., Yan, C., Soomro, S. E. H., Zafar, M. A., Li, Z., Xu, J., & Yousaf, A. (2023). Two-Dimensional Hydrodynamic Modeling for Prediction of Bank Erosion and Bed Incision in the Indus River. *Acta Geophysica*, 1-18.
- Borkin, D., Némethová, A., Michal'čonok, G., & Maiorov, K. (2019). Impact of Data Normalization on Classification Model Accuracy. *Research Papers Faculty of Materials Science and Technology Slovak University of Technology*, 27(45), 79-84.
- Bustami, R., Bessaih, N., Bong, C., & Suhaili, S. (2007). Artificial Neural Network for Precipitation and Water Level Predictions of Bedup River. *IAENG International Journal of Computer Science*, 34(2).

- Castrounis, A. (2016). Artificial Intelligence, Deep Learning, and Neural Networks, Explained. *KDnuggets Informatics Blog*. [online] Available at: <https://www.kdnuggets.com/2016/10/artificial-intelligence-deep-learning-neural-networks-explained.html> [Accessed on 10 April 2023]
- Chang, M. J., Chang, H. K., Chen, Y. C., Lin, G. F., Chen, P. A., Lai, J. S., & Tan, Y. C. (2018). A Support Vector Machine Forecasting Model for Typhoon Flood Inundation Mapping and Early Flood Warning Systems. *Water*, 10(12), 1734.
- Chen, J., Li, Y., Zhang, C., Tian, Y., & Guo, Z. (2023). Urban Flooding Prediction Method Based on the Combination of LSTM Neural Network and Numerical Model. *International Journal of Environmental Research and Public Health*, 20(2), 1043.
- Chen, H., Zhang, Y., Kalra, M. K., Lin, F., Chen, Y., Liao, P., ... & Wang, G. (2017). Low-dose CT with a Residual Encoder-Decoder Convolutional Neural Network. *IEEE Transactions on Medical Imaging*, 36(12), 2524-2535.
- Chitra, P., & Rajasekaran, U. M. (2022). Time-Series Analysis and Flood Prediction using a Deep Learning Approach. In *Proceedings of IEEE 2022 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET)*, (pp. 139-142).
- Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv preprint arXiv:1406.1078*.
- Chu, H., Wu, W., Wang, Q. J., Nathan, R., & Wei, J. (2020). An ANN-based Emulation Modelling Framework for Flood Inundation Modelling: Application, Challenges and Future Directions. *Environmental Modelling & Software*, 124, 104587.

- D'Addabbo, A., Refice, A., Pasquariello, G., Lovergine, F. P., Capolongo, D., & Manfreda, S. (2016). A Bayesian Network for Flood Detection Combining Sar Imagery and Ancillary Data. In *Proceedings of IEEE Transactions on Geoscience and Remote Sensing*, 54(6), (pp.3612-3625).
- Dai, W., & Cai, Z. (2021). Predicting Coastal Urban Floods using Artificial Neural Network: The Case Study of Macau, China. *Applied Water Science*, 11(10), 161.
- Damavandi, H. G., Shah, R., Stampoulis, D., Wei, Y., Bosovic, D., & Sabo, J. (2019). Accurate Prediction of Streamflow using Long Short-Term Memory Network: A Case Study in the Brazos River Basin In Texas. *International Journal of Environmental Science and Development*, 10(10), 294-300.
- Dandot, D. W. B., Ismuni, I. H., Murang, O., Jones, D. J., Siaw, C., Dari, C., Othman, H., ... Ming, L. W. (2006). *Development in Sarawak an Aerial Perspective: Miri* (2nd ed.), pp. 173-208. Sarawak, Malaysia: Osacar Sdn Bhd, Malaysia.
- Dash, P., & Sar, J. (2020). Identification and Validation of Potential Flood Hazard Area using GIS-based Multi-Criteria Analysis and Satellite Data-Derived Water Index. *Journal of Flood Risk Management*, 13(3), e12620.
- Dehghani, M., Saghafian, B., Nasiri Saleh, F., Farokhnia, A., & Noori, R. (2014). Uncertainty Analysis of Streamflow Drought Forecast using Artificial Neural Networks and Monte-Carlo Simulation. *International Journal of Climatology*, 34(4), 1169-1180.
- Department of Irrigation and Drainage Sarawak. (2018). *Hydrology Stations at Baram Basin*. [online] Available at: https://did.sarawak.gov.my/web/subpage/webpage_view/333 [Accessed on 10 April 2023]

- Dibike, Y. B., Velickov, S., Solomatine, D., & Abbott, M. B. (2001). Model Induction with Support Vector Machines: Introduction and Applications. *Journal of Computing in Civil Engineering*, 15(3), 208-216.
- Ding, Y., Zhu, Y., Feng, J., Zhang, P., & Cheng, Z. (2020). Interpretable Spatio-Temporal Attention LSTM Model for Flood Forecasting. *Neurocomputing*, 403, 348-359.
- Di Nunno, F., de Marinis, G., & Granata, F. (2023). Short-Term Forecasts of Streamflow in the UK based on a Novel Hybrid Artificial Intelligence Algorithm. *Scientific Reports*, 13(1), 7036.
- Duan, Y., Yisheng, L. V., & Wang, F. Y. (2016). Travel Time Prediction with LSTM Neural Network. In *Proceedings of IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, (pp. 1053-1058).
- Dubey, A. D. (2015). Artificial Neural Network Models for Rainfall Prediction in Pondicherry. *International Journal of Computer Applications*, 120(3), 30-35.
- Ehteram, M., & Ghanbari-Adivi, E. (2023). Self-Attention (SA) Temporal Convolutional Network (SATCN)-Long Short-Term Memory Neural Network (SATCN-LSTM): An Advanced Python Code for Predicting Groundwater Level. *Environmental Science and Pollution Research*, 1-19.
- Elias, Z., Hamin, Z., & Othman, M. B. (2013). Sustainable Management of Flood Risks in Malaysia: Some Lessons from the Legislation in England and Wales. *Procedia-Social and Behavioral Sciences*, 105, 491-497.
- Fang, W., Yu, Z., Chen, Y., Huang, T., Masquelier, T., & Tian, Y. (2021). Deep Residual Learning in Spiking Neural Networks. *Advances in Neural Information Processing Systems*, 34, 21056-21069.

- Faudzi, A. A. M., Raslan, M. M., & Alias, N. E. (2023). IoT based Real-Time Monitoring System of Rainfall and Water Level for Flood Prediction using LSTM Network. In *Proceedings of IOP Conference Series: Earth and Environmental Science*, 1143(1), (pp. 012015). IOP Publishing.
- Fawcett, T. (2006). An Introduction to ROC Analysis. *Pattern Recognition Letters*, 27(8), 861-874.
- Fenglin, W., Ahmad, I., Zelenakova, M., Fenta, A., Dar, M. A., Teka, A. H., ... & Shafi, S. N. (2023). Exploratory Regression Modeling for Flood Susceptibility Mapping in the GIS Environment. *Scientific Reports*, 13(1), 247.
- Fitzpatrick, L., Titze, D., Anderson, E. J., Beletsky, D., & Kelley, J. G. (2023). Simulating Flood Events at the Twin Ports of Duluth-Superior using a Linked Hydrologic-Hydrodynamic Framework. *Ocean Dynamics*, 1-15.
- Fotovatikhah, F., Herrera, M., Shamshirband, S., Chau, K. W., Faizollahzadeh Ardabili, S., & Piran, M. J. (2018). Survey of Computational Intelligence as Basis to Big Flood Management: Challenges, Research Directions and Future Work. *Engineering Applications of Computational Fluid Mechanics*, 12(1), 411-437.
- Fukushima, K., & Miyake, S. (1982). Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Visual Pattern Recognition. Competition and Cooperation in Neural Nets. In *Proceedings of the US-Japan Joint Seminar*, (pp. 267-85).
- Gao, L., Zhang, L., Hong, Y., Chen, H. X., & Feng, S. J. (2023). Flood Hazards in Urban Environment. *Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards*, 17(2), 241-261.

- Gallego, G., Delbrück, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., ... & Scaramuzza, D. (2020). Event-Based Vision: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1), 154-180.
- Gautrais, J., & Thorpe, S. (1998). Rate Coding Versus Temporal Order Coding: A Theoretical Approach. *Biosystems*, 48(1-3), 57-65.
- Geminiani, A., Pedrocchi, A., D'Angelo, E., & Casellato, C. (2019). Response Dynamics in an Olivocerebellar Spiking Neural Network with Non-Linear Neuron Properties. *Frontiers in Computational Neuroscience*, 13, 68.
- Gers, F. A., Schraudolph, N. N., & Schmidhuber, J. (2002). Learning Precise Timing with LSTM Recurrent Networks. *Journal of Machine Learning Research*, 3(8), 115-143.
- Gerstner, W., Kistler, W. M., Naud, R., & Paninski, L. (2014). *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition*, Cambridge University Press.
- Graves, A., & Jaitly, N. (2014). Towards End-To-End Speech Recognition with Recurrent Neural Networks. In *Proceedings of 31st International Conference on Machine Learning* 32(2), (pp. 1764-1772). PMLR.
- Gray, C. M., & Singer, W. (1989). Stimulus-Specific Neuronal Oscillations in Orientation Columns of Cat Visual Cortex. In *Proceedings of the National Academy of Sciences*, 86(5), (pp.1698-1702).
- Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2016). LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10), 2222-2232.

- Guo, Y., & Lai, X. J. (2020). Water Level Prediction of Lake Poyang Based on Long Short-Term Memory Neural Network. *Journal of Lake Sciences*, 32(3), 865-876.
- Hansana, P., Guo, X., Zhang, S., Kang, X., & Li, S. (2023). Flood Analysis using Multi-Scale Remote Sensing Observations in Laos. *Remote Sensing*, 15(12), 3166.
- Hashemi, R., Brigode, P., Garambois, P. A., & Javelle, P. (2022). How can We Benefit from Regime Information to make more Effective use of Long Short-Term Memory (LSTM) Runoff Models? *Hydrology and Earth System Sciences*, 26(22), 5793-5816.
- He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on Imagenet Classification. In *Proceedings of the IEEE International Conference on Computer Vision*, (pp. 1026-1034).
- Hinton, G. E. (2007). Learning Multiple Layers of Representation. *Trends in Cognitive Sciences*, 11(10), 428-434.
- Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, 18(7), 1527-1554.
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735-1780.
- Hodgkin, A. L., & Huxley, A. F. (1952). Currents Carried by Sodium and Potassium Ions Through the Membrane of the Giant Axon of Loligo. *The Journal of Physiology*, 116(4), 449.

- Hosein, S., & Hosein, P. (2017). Load Forecasting using Deep Neural Networks. In *Proceedings of 2017 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, (pp. 1-5).
- Hu, Y., Yan, L., Hang, T., & Feng, J. (2020). Stream-Flow Forecasting of Small Rivers based on LSTM. *arXiv preprint arXiv:2001.05681*.
- Huang, H., Lei, X., Liao, W., Liu, D., & Wang, H. (2023). A Hydrodynamic-Machine Learning Coupled (HMC) Model of Real-Time Urban Flood in a Seasonal River Basin using Mechanism-Assisted Temporal Cross-Correlation (MTC) for Space Decoupling. *Journal of Hydrology*, 129826.
- Hubel, D. H., & Wiesel, T. N. (1962). Receptive Fields, Binocular Interaction and Functional Architecture in the Cat's Visual Cortex. *The Journal of Physiology*, 160(1), 106.
- Ian, V. K., Tse, R., Tang, S. K., & Pau, G. (2023). Bridging the Gap: Enhancing Storm Surge Prediction and Decision Support with Bidirectional Attention-Based LSTM. *Atmosphere*, 14(7), 1082.
- Ighile, E. H., Shirakawa, H., & Tanikawa, H. (2022). Application of GIS and Machine Learning to Predict Flood Areas in Nigeria. *Sustainability*, 14(9), 5039.
- Ignatov, M., Ziegler, M., Hansen, M., Petraru, A., & Kohlstedt, H. (2015). A Memristive Spiking Neuron with Firing Rate Coding. *Frontiers in Neuroscience*, 9, 376.
- Ikeda, S., & Manton, J. H. (2009). Capacity of A Single Spiking Neuron. In *proceedings of Journal of Physics: Conference Series*, 197(1), (pp. 012014). IOP Publishing.
- Jajarmizadeh, M., Harun, S., & Salarpour, M. (2014). An Assessment of a Proposed Hybrid Neural Network for Daily Flow Prediction in Arid Climate. *Modelling and Simulation in Engineering*, 2014, 1-10.

- Jin, B., Che, C., Liu, Z., Zhang, S., Yin, X., & Wei, X. (2018). Predicting the Risk of heart Failure with EHR Sequential Data Modeling. *IEEE Access*, 6, 9256-9261.
- Johansson, R. S., & Birznieks, I. (2004). First Spikes in Ensembles of Human Tactile Afferents Code Complex Spatial Fingertip Events. *Nature Neuroscience*, 7(2), 170-177.
- Karim, F., Armin, M. A., Ahmedt-Aristizabal, D., Tychsen-Smith, L., & Petersson, L. (2023). A Review of Hydrodynamic and Machine Learning Approaches for Flood Inundation Modeling. *Water*, 15(3), 566.
- Karpathy, A., & Fei-Fei, L. (2015). Deep Visual-Semantic Alignments for Generating Image Descriptions. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 3128-3137).
- Kasyanto, H., Sari, R. R., & Lubis, M. F. (2023). Application of Multilayer Perceptron (MLP) Method for Streamflow Forecasting (Case Study: Upper Citarum River, Indonesia). In *Proceedings of IOP Conference Series: Earth and Environmental Science*, 1203(1), (pp. 012032). IOP Publishing.
- Ke, Z., & Vikalo, H. (2021). Real-Time Radio Technology and Modulation Classification Via an LSTM Auto-Encoder. *IEEE Transactions on Wireless Communications*, 21(1), 370-382.
- Keong, K. C., Mustafa, M., Mohammad, A. J., Sulaiman, M. H., Abdullah, N. R. H., Samad, R., & Pebrianti, D. (2016). Levenberg-Marquardt Flood Prediction for Sungai Isap Residence. In *Proceedings of 2016 IEEE Conference on Systems, Process and Control (ICSPC)*, (pp. 160-165).

- Kessler, F. L., & Jong, J. (2015). Incision of Rivers in Pleistocene Gravel and Conglomeratic Terraces: Further Circumstantial Evidence for the Uplift Of Borneo During The Neogene and Quaternary. *Bulletin of the Geological Society of Malaysia*, 61, 49-57.
- Khalaf, M., Hussain, A. J., Al-Jumeily, D., Fergus, P., & Idowu, I. O. (2015). Advance Flood Detection and Notification System Based on Sensor Technology and Machine Learning Algorithm. In *Proceedings of IEEE 2015 International Conference on Systems, Signals and Image Processing (IWSSIP)*, (pp. 105-108).
- Kim, D., Park, J., Han, H., Lee, H., Kim, H. S., & Kim, S. (2023). Application of AI-Based Models for Flood Water Level Forecasting and Flood Risk Classification. *KSCE Journal of Civil Engineering*, 27, 3163–3174.
- Kong, Y. L., Huang, Q., Wang, C., Chen, J., Chen, J., & He, D. (2018). Long Short-Term Memory Neural Networks for Online Disturbance Detection in Satellite Image Time Series. *Remote Sensing*, 10(3), 452.
- Kourgialas, N. N., & Karatzas, G. P. (2011). Flood Management and A GIS Modelling Method to Assess Flood-Hazard Areas: A Case Study. *Hydrological Sciences Journal*, 56(2), 212-225.
- Kratzert, F., Klotz, D., Brenner, C., Schulz, K., & Herrnegger, M. (2018). Rainfall Runoff Modelling using Long Short-Term Memory (LSTM) Networks. *Hydrology and Earth System Sciences*, 22(11), 6005-6022.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, 25.

- Krstinic, D., Braovic, M., Seric, L., & Bozic-Stulic, D. (2020). Multi-Label Classifier Performance Evaluation with Confusion Matrix. *Computer Science & Information Technology, 1*.
- Kumar, V., Azamathulla, H. M., Sharma, K. V., Mehta, D. J., & Maharaj, K. T. (2023a). The State of the Art in Deep Learning Applications, Challenges, and Future Prospects: A Comprehensive Review of Flood Forecasting and Management. *Sustainability, 15*(13), 10543.
- Kumar, V., Sharma, K. V., Caloiero, T., Mehta, D. J., & Singh, K. (2023b). Comprehensive Overview of Flood Modeling Approaches: A Review of Recent Advances. *Hydrology, 10*(7), 141.
- Kuok, K. K., Harun, S., & Chan, C. P. (2011). Hourly Runoff Forecast at Different Leadtime for a Small Watershed using Artificial Neural Networks. *International Journal of Advances in Soft Computing and Its Applications, 3*(1), 68-86.
- Kuok, K. K., Harun, S., & Shamsuddin, S. M. (2010). Particle Swarm Optimization Feedforward Neural Network for Hourly Rainfall-Runoff Modeling in Bedup Basin, Malaysia. *International Journal of Civil & Environmental Engineering, 9*(10), 9-18.
- Labatut, V., & Cherifi, H. (2011). Evaluation of Performance Measures for Classifiers Comparison. *arXiv preprint arXiv:1112.4133*.
- Lai, C. (2023, February 17). Sudden Downpour Causes Flash Floods along Miri Roads. *Borneo Post*. [online] Available at:
<https://www.theborneopost.com/2023/02/17/sudden-downpour-causes-flash-floods-along-miri-roads/> [Accessed on 9 April 2023]

- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. *Nature*, 521(7553), 436-444.
- Le, Q. V. (2013). Building High-Level Features using Large Scale Unsupervised Learning. In *Proceedings of 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, (pp. 8595-8598).
- Li, Q., Song, D., Yuan, C., & Nie, W. (2022). An Image Recognition Method for the Deformation Area of Open-Pit Rock Slopes under Variable Rainfall. *Measurement*, 188, 110544.
- Li, X., & Willems, P. (2020). A Hybrid Model for Fast and Probabilistic Urban Pluvial Flood Prediction. *Water Resources Research*, 56(6), e2019WR025128.
- Liang, C., Li, H., Lei, M., & Du, Q. (2018). Dongting Lake Water Level Forecast and its Relationship with the Three Gorges Dam Based on a Long Short-Term Memory Network. *Water*, 10(10), 1389.
- Lichtsteiner, P., Posch, C., & Delbruck, T. (2006). A 128 x 128 120db 30mw Asynchronous Vision Sensor that Responds to Relative Intensity Change. In *Proceedings of 2006 IEEE International Solid State Circuits Conference-Digest of Technical Papers*, (pp. 2060-2069).
- Lin, Q., Leandro, J., Gerber, S., & Disse, M. (2020). Multistep Flood Inundation Forecasts With Resilient Backpropagation Neural Networks: Kulmbach Case Study. *Water*, 12(12), 3568.
- Ling, T. Y., Soo, C. L., Liew, J. J., Nyanti, L., Sim, S. F., & Grinang, J. (2017). Application of Multivariate Statistical Analysis in Evaluation of Surface River Water Quality of a Tropical River. *Journal of Chemistry*, 2017, 1-13.

- Luo, P., Luo, M., Li, F., Qi, X., Huo, A., Wang, Z., ... & Nover, D. (2022). Urban Flood Numerical Simulation: Research, Methods and Future Perspectives. *Environmental Modelling & Software*, 156, 105478.
- Liu, M., Huang, Y., Li, Z., Tong, B., Liu, Z., Sun, M., ... & Zhang, H. (2020). The Applicability of LSTM-KNN Model for Real-Time Flood Forecasting in Different Climate Zones in China. *Water*, 12(2), 440.
- Liu, F., Xu, F., & Yang, S. (2017). A Flood Forecasting Model Based on Deep Learning Algorithm via Integrating Stacked Autoencoders with BP Neural Network. In *Proceedings of 2017 IEEE 3rd International Conference on Multimedia Big Data (BigMM)*, (pp. 58-61).
- Liu, S. C., Van Schaik, A., Minch, B. A., & Delbruck, T. (2010). Event-Based 64-Channel Binaural Silicon Cochlea with Q Enhancement Mechanisms. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, (pp. 2027-2030).
- Maass, W. (1997). Networks of Spiking Neurons: The Third Generation of Neural Network Models. *Neural Networks*, 10(9), 1659-1671.
- Maier, H. R., Jain, A., Dandy, G. C., & Sudheer, K. P. (2010). Methods used for The Development of Neural Networks For The Prediction of Water Resource Variables in River Systems: Current Status and Future Directions. *Environmental Modelling & Software*, 25(8), 891-909.
- Marino, D. L., Amarasinghe, K., & Manic, M. (2016). Building Energy Load Forecasting using Deep Neural Networks. In *Proceedings of IECON 2016 42nd Annual Conference of the IEEE Industrial Electronics Society*, (pp. 7046-7051).

- Mirzavand, M., & Ghazavi, R. (2015). A Stochastic Modelling Technique for Groundwater Level Forecasting in an Arid Environment using Time Series Methods. *Water Resources Management*, 29, 1315-1328.
- Moeeni, H., & Bonakdari, H. (2017). Forecasting Monthly Inflow with Extreme Seasonal Variation using the Hybrid SARIMA-ANN Model. *Stochastic Environmental Research and Risk Assessment*, 31, 1997-2010.
- Mohammed, N., Edwards, R., & Gale, A. (2018). Optimisation of Flooding Recovery for Malaysian Universities. *Procedia Engineering*, 212, 356-362.
- Moishin, M., Deo, R. C., Prasad, R., Raj, N., & Abdulla, S. (2021). Designing Deep-Based Learning Flood Forecast Model with ConvLSTM Hybrid Algorithm. *IEEE Access*, 9, 50982-50993.
- Moreno-Torres, J. G., Sáez, J. A., & Herrera, F. (2012). Study on the Impact of Partition-Induced Dataset Shift on k -fold Cross-Validation. *IEEE Transactions on Neural Networks and Learning Systems*, 23(8), 1304-1312.
- Mosavi, A., Ozturk, P., & Chau, K. W. (2018). Flood Prediction using Machine Learning Models: Literature Review. *Water*, 10(11), 1536.
- Mousavi, F. S., Yousefi, S., Abghari, H., & Ghasemzadeh, A. (2021). Design of an IoT-based Flood Early Detection System using Machine Learning. In *Proceedings of 2021 IEEE 26th International Computer Conference, Computer Society of Iran, (CSICC)*, (pp. 1-7).
- Nie, W., Krautblatter, M., Leith, K., Thuro, K., & Festl, J. (2017). A Modified Tank Model Including Snowmelt and Infiltration Time Lags for Deep-Seated Landslides in Alpine Environments (Aggenalm, Germany). *Natural Hazards and Earth System Sciences*, 17(9), 1595-1610.

- Oh, B. D., Song, H. J., Kim, J. D., Park, C. Y., & Kim, Y. S. (2019). Predicting Concentration of PM10 Using Optimal Parameters of Deep Neural Network. *Intelligent Automation & Soft Computing*, 25(2), 343-350.
- Othman, M., Ahmad, M. N., Suliman, A., Arshad, N. H., & Maidin, S. S. (2014). COBIT Principles to Govern Flood Management. *International Journal of Disaster Risk Reduction*, 9, 212-223.
- Pal, K., & Patel, B. V. (2020). Data Classification with k -Fold Cross Validation and Holdout Accuracy Estimation Methods with 5 Different Machine Learning Techniques. In *Proceedings of IEEE 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, (pp. 83-87).
- Panahi, M., Dodangeh, E., Rezaie, F., Khosravi, K., Van Le, H., Lee, M. J., ... & Pham, B. T. (2021). Flood Spatial Prediction Modeling using a Hybrid of Meta-Optimization and Support Vector Regression Modeling. *Catena*, 199, 105114.
- Paul, A., & Das, P. (2014). Flood Prediction Model Using Artificial Neural Network. *International Journal of Computer Applications Technology and Research*, 3(7), 473-478.
- Pei, J., Deng, L., Song, S., Zhao, M., Zhang, Y., Wu, S., ... & Shi, L. (2019). Towards Artificial General Intelligence with Hybrid Tianjic Chip Architecture. *Nature*, 572(7767), 106-111.
- Petro, B., Kasabov, N., & Kiss, R. M. (2019). Selection and Optimization of Temporal Spike Encoding Methods for Spiking Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 31(2), 358-370.

- Phitakwinai, S., Auephanwiriyaikul, S., & Theera-Umpon, N. (2016). Multilayer Perceptron with Cuckoo Search in Water Level Prediction for Flood Forecasting. In *Proceedings of 2016 IEEE International Joint Conference on Neural Networks (IJCNN)*, (pp. 519-524).
- Pirah, J., & Roslee, R. (2021). Positive Changes in Flood Mitigation Through Sand Dredging Works at Padas River and Tributary Based on Hec-Ras Hydrological Modelling. *Journal of Design & Nature and Ecodynamics*, 16(4), 451-458.
- Poornima, S., & Pushpalatha, M. (2019). Prediction of Rainfall Using Intensified LSTM based Recurrent Neural Network with Weighted Linear Units. *Atmosphere*, 10(11), 668.
- Ratnasari, A. D., Hasin, K., Soim, S., Setiawan, B. T., & Arfianto, A. Z. (2022). Flood Detection using Backpropagation Neural Network Method. *Indonesian Journal of Engineering Research*, 3(1), 1-10.
- Rezaeianzadeh, M., Tabari, H., Arabi Yazdi, A., Isik, S., & Kalin, L. (2014). Flood Flow Forecasting using ANN, ANFIS and Regression Models. *Neural Computing and Applications*, 25, 25-37.
- Riad, S., Mania, J., Bouchaou, L., & Najjar, Y. (2004). Rainfall-Runoff Model Using an Artificial Neural Network Approach. *Mathematical and Computer Modelling*, 40(7-8), 839-846.
- Ringgit, M. (2022, December 26). Districts in Miri Affected by Floods. *New Sarawak Tribune*. [online] Available at:
<https://www.nat:rawaktribune.com.my/districts-in-miri-affected-by-floods/>
 [Accessed on 9 April 2023]

- Roondiwala, M., Patel, H., & Varma, S. (2017). Predicting Stock Prices Using LSTM. *International Journal of Science and Research (IJSR)*, 6(4), 1754-1756.
- Roji Kawi, M. (2023, March 18). Five Floods in Two Months Leave Kuching Folk Frustrated. *New Straits Times*. [online] Available at: <https://www.nst.com.my/news/nation/2023/03/890401/five-floods-two-months-leave-kuching-folk-frustrated> [Accessed on 9 April 2023]
- Roy, K., Jaiswal, A., & Panda, P. (2019). Towards Spike-Based Machine Intelligence with Neuromorphic Computing. *Nature*, 575(7784), 607-617.
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning Representations by Back-Propagating Errors. *Nature*, 323(6088), 533-536.
- Ruslan, F. A., Tajuddin, M., & Adnan, R. (2015). Flood Prediction Modeling Using Improved MLPNN Structure: Case Study Kuala Lumpur. In *Proceedings of 2015 IEEE Conference on Systems, Process and Control (ICSPC)*, (pp. 101-105).
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Fei-Fei, L. (2015). Imagenet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115, 211-252.
- Safiah Yusmah, M. Y., Bracken, L. J., Sahdan, Z., Norhaslina, H., Melasutra, M. D., Ghaffarianhoseini, A., ... & Shereen Farisha, A. S. (2020). Understanding Urban Flood Vulnerability and Resilience: A Case Study pf Kuantan, Pahang, Malaysia. *Natural Hazards*, 101, 551-571.
- Saha, M., Mitra, P., & Nanjundiah, R. S. (2017). Deep Learning for Predicting the Monsoon over the Homogeneous Regions of India. *Journal of Earth System Science*, 126, 1-18.

- Saha, M., & Mitra, P. (2016). Recurrent Neural Network based Prediction of Indian Summer Monsoon using Global Climatic Predictors. In *Proceedings of 2016 IEEE International Joint Conference on Neural Networks (IJCNN)*, (pp. 1523-1529).
- Sahoo, A., & Ghose, D. K. (2022). Application of Hybrid Support Vector Machine Model for Streamflow Prediction in Barak Valley, India. In *Proceedings of IOP Conference Series: Earth and Environmental Science*, 1032(1), (pp.012016). IOP Publishing.
- Salleh, N. A., Mustaffa, C. S., & Mohd Taib, A. (2013). Proposing Instrument to Measure Impression Management Among Flood Victims. *International Journal of Social Science and Humanity*, 3(6), 538-542.
- Salman, A. G., Kanigoro, B., & Heryadi, Y. (2015). Weather Forecasting Using Deep Learning Techniques. In *Proceedings of 2015 IEEE International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, (pp. 281-285).
- Samual, A. (2023, March 18). Flash Floods Strike Kuching City, Kota Samarahan. *Borneo Post*. [online] Available at: <https://www.theborneopost.com/2023/03/18/flash-floods-strike-kuching-city-kota-samarahan/> [Accessed on 9 April 2023]
- Sarawak River Board. (2018). *Batang Baram Anchorage Area*. [online] Available at: <https://srb.sarawak.gov.my/page-0-0-133-Batang-Baram-Anchorage-Area.html> [Accessed on 9 April 2020]
- Schmidhuber, J. (2015). Deep Learning in Neural Networks: An Overview. *Neural Networks*, 61, 85-117.
- See, L., & Openshaw, S. (2000). A Hybrid Multi-Model Approach to River Level Forecasting. *Hydrological Sciences Journal*, 45(4), 523-536.

- Selvin, S., Vinayakumar, R., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. (2017). Stock Price Prediction Using LSTM, RNN and CNN-Sliding Window Model. In *Proceedings of 2017 IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, (pp. 1643-1647).
- Senthil Kumar, A. R., Sudheer, K. P., Jain, S. K., & Agarwal, P. K. (2005). Rainfall-Runoff Modelling using Artificial Neural Networks: Comparison of Network Types. *Hydrological Processes: An International Journal*, 19(6), 1277-1291.
- Shi, Y., Taalab, K., & Cheng, T. (2016). Flood Prediction Using Support Vector Machines (SVM). In *Proceeding of 24th GIS Research UK (GISRUK) Conference (GISRUK2016)*, (pp.1-9).
- Shuofeng, L., Puwen, L., & Koyamada, K. (2021). LSTM based Hybrid Method for Basin Water Level Prediction by Using Precipitation Data. *Journal of Advanced Simulation in Science and Engineering*, 8(1), 40-52.
- Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556*.
- Singh, S., & Kasana, S. S. (2022). Quantitative Estimation of Soil Properties Using Hybrid Features and RNN Variants. *Chemosphere*, 287, 131889.
- Skrobek, D., Krzywanski, J., Sosnowski, M., Kulakowska, A., Zylka, A., Grabowska, K., ... & Nowak, W. (2020). Prediction of Sorption Processes Using the Deep Learning Methods (Long Short-Term Memory). *Energies*, 13(24), 6601.
- Soomlek, C., Kaewchainam, N., Simano, T., & So-In, C. (2015). Using Backpropagation Neural Networks for Flood Forecasting in PhraNakhon Si Ayutthaya, Thailand. In *Proceedings of 2015 IEEE International Computer Science and Engineering Conference (ICSEC)*, (pp. 1-6).

- Song, T., Ding, W., Wu, J., Liu, H., Zhou, H., & Chu, J. (2019). Flash Flood Forecasting based on Long Short-Term Memory Networks. *Water*, 12(1), 109.
- Song, B., Fan, C., Wu, Y., & Sun, J. (2018). Data Prediction for Public Events in Professional Domains based on Improved RNN-LSTM. In *Proceedings of IOP Journal of Physics: Conference Series*, 976(1), (pp. 012007). IOP Publishing.
- Stephen, T. (2018, February 8). The Number of Flood Evacuees in Sarawak Rises to more than 5,000. *The Star Online. Miri*. [online] Available at: <https://www.thestar.com.my/news/nation/2018/02/08/floods-in-sarawak/> [Accessed on 9 November 2020]
- Stephen, T. (2017, December 20). Flash Floods Hit Northern Sarawak Following Incessant Rain. *The Star Online. Miri*. [online] Available at: <https://www.thestar.com.my/news/nation/2017/12/20/flash-floods-hit-northern-sarawak-following-incessant-rain/> [Accessed on 9 November 2020]
- Sugiartawan, P., Pulungan, R., & Sari, A. K. (2017). Prediction by a Hybrid of Wavelet Transform and long-Short-Term-Memory Neural Network. *International Journal of Advanced Computer Science and Applications*, 8(2), 326-332.
- Sulaiman, J., & Wahab, S. H. (2018). Heavy Rainfall Forecasting Model Using Artificial Neural Network for Flood Prone Area. In *proceedings of IT Convergence and Security 2017*(1), (pp. 68-76). Springer Singapore.
- Sun, H., Dai, X., Shou, W., Wang, J., & Ruan, X. (2021). An Efficient Decision Support System for Flood Inundation Management using Intermittent Remote-Sensing Data. *Remote Sensing*, 13(14), 2818.

- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going Deeper with Convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 1-9).
- Tan, M. L., Ibrahim, A. L., Duan, Z., Cracknell, A. P., & Chaplot, V. (2015). Evaluation of Six High-Resolution Satellite and Ground-Based Precipitation Products over Malaysia. *Remote Sensing*, 7(2), 1504-1528.
- Teng, J., Jakeman, A. J., Vaze, J., Croke, B. F., Dutta, D., & Kim, S. J. E. M. (2017). Flood Inundation Modelling: A Review of Methods, Recent Advances and Uncertainty Analysis. *Environmental Modelling & Software*, 90, 201-216.
- Tew, Y. L., Tan, M. L., Juneng, L., Chun, K. P., Hassan, M. H. B., Osman, S. B., ... & Kabir, M. H. (2022). Rapid Extreme Tropical Precipitation and Flood Inundation Mapping Framework (RETRACE): Initial Testing for the 2021–2022 Malaysia Flood. *ISPRS International Journal of Geo-Information*, 11(7), 378.
- Thamilselvan, R., Prabhu, M., Selvi, K. T., Karthik, V., & Bhuvaneshwaran, T. (2022). An Extensive Review of Rainfall Prediction using Machine Learning and Deep Learning Techniques. In *Proceedings of 2022 IEEE Second International Conference on Artificial Intelligence and Smart Energy (ICAIS)*, (pp. 198-205).
- The Official Web of Public Infobanjir. (2018). *Categorization of Rainfall Intensity*. Ministry of Natural Resources, Environment and Climate Change. [online] Available at: <https://publicinfobanjir.water.gov.my/hujan/data-hujan/?lang=en> [Accessed on 9 November 2020]

- Theilen-Willige, B., & Budiyanto, E. (2022). Flooding Hazard Assessment Considering Climate Change in the Coastal Areas of Java/Indonesia Based on Remote Sensing and GIS Data. In *Proceedings of International Joint Conference on Arts and Humanities 2021 (IJCAH 2021)*, (pp. 682-690). Atlantis Press.
- Timbadiya, P. V., & Krishnamraju, K. M. (2023). A 2D Hydrodynamic Model for River Flood Prediction in a Coastal Floodplain. *Natural Hazards*, 115(2), 1143-1165.
- Tillihal, S. B., & Shukla, A. K. (2023). Flood Disaster Hazards: A State-of-the-Art Review of Causes, Impacts, and Monitoring. *Advances in Water Resource Planning and Sustainability*, 77-95.
- Tsakiri, K., Marsellos, A., & Kapetanakis, S. (2018). Artificial Neural Network and Multiple Linear Regression for Flood Prediction in Mohawk River, New York. *Water*, 10(9), 1158.
- Valipour, M., Banihabib, M. E., & Behbahani, S. M. R. (2013). Comparison of the ARMA, ARIMA, and the Autoregressive Artificial Neural Network Models in Forecasting the Monthly Inflow of Dez Dam Reservoir. *Journal of Hydrology*, 476, 433-441.
- Vapnik, V., & Mukherjee, S. (1999). Support Vector Method for Multivariate Density Estimation. *Advances in Neural Information Processing Systems*, 12, 1-7.
- Viteri López, A. S., & Morales Rodriguez, C. A. (2020). Flash Flood Forecasting in Sao Paulo Using a Binary Logistic Regression Model. *Atmosphere*, 11(5), 473.
- Von Der Malsburg, C. (1994). The Correlation Theory of Brain Function. In *Models of Neural Networks: Temporal Aspects of Coding and Information Processing in Biological Systems*, pp. 95-119. New York, NY: Springer New York.

- Wang, Y., Huang, Y., Xiao, M., Zhou, S., Xiong, B., & Jin, Z. (2023). Medium Long-Term Prediction of Water Level based on an Improved Spatio-Temporal Attention Mechanism for Long Short-Term Memory Networks. *Journal of Hydrology*, 618, 129163.
- Widiasari, I. R., Nugoho, L. E., & Efendi, R. (2018). Context-based Hydrology Time Series Data for a Flood Prediction Model Using LSTM. In *Proceedings of 2018 IEEE 5th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, (pp. 385-390).
- Wu, W., Emerton, R., Duan, Q., Wood, A. W., Wetterhall, F., & Robertson, D. E. (2020). Ensemble Flood Forecasting: Current Status and Future Opportunities. *Wiley Interdisciplinary Reviews: Water*, 7(3), e1432.
- Wu, Z., Zhang, H., Lin, Y., Li, G., Wang, M., & Tang, Y. (2021). LIAF-NET: Leaky Integrate and Analog Fire Network for Lightweight and Efficient Spatiotemporal Information Processing. *IEEE Transactions on Neural Networks and Learning Systems*, 33(11), 6249-6262.
- Xiang, Z., Yan, J., & Demir, I. (2020). A Rainfall-Runoff Model with LSTM-based Sequence-to-Sequence Learning. *Water Resources Research*, 56(1), e2019WR025326.
- Xie, Z., Liu, Q., & Cao, Y. (2021). Hybrid Deep Learning Modeling for Water Level Prediction in Yangtze River. *Intelligent Automation & Soft Computing*, 28(1), 153-166.
- Xu, Y., Hu, C., Wu, Q., Li, Z., Jian, S., & Chen, Y. (2021). Application of Temporal Convolutional Network for Flood Forecasting. *Hydrology Research*, 52(6), 1455-1468.

- Yadav, S., & Shukla, S. (2016). Analysis of k-fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification. In *Proceedings of 2016 IEEE 6th International Conference on Advanced Computing (IACC)*, (pp. 78-83).
- Yan, L., Chen, C., Hang, T., & Hu, Y. (2021). A Stream Prediction Model Based on Attention-LSTM. *Earth Science Informatics*, 14, 723-733.
- Yan, J., Jin, J., Chen, F., Yu, G., Yin, H., & Wang, W. (2018). Urban Flash Flood Forecast using Support Vector Machine and Numerical Simulation. *Journal of Hydroinformatics*, 20(1), 221-231.
- Yang, C. H., Wu, C. H., & Hsieh, C. M. (2020). Long Short-Term Memory Recurrent Neural Network for Tidal Level Forecasting. *IEEE Access*, 8, 159389-159401.
- Yang, H., Pan, Z., & Tao, Q. (2017). Robust and Adaptive Online Time Series Prediction with Long Short-Term Memory. *Computational Intelligence and Neuroscience*, 2017, 9478952.
- Yap, Z. N., & Musa, S. (2023). Stream Flow Forecasting on Pahang River by Time Series Models, ARMA, ARIMA and SARIMA. *Recent Trends in Civil Engineering and Built Environment*, 4(1), 331-341.
- Yariyan, P., Avand, M., Abbaspour, R. A., Torabi Haghighi, A., Costache, R., Ghorbanzadeh, O., ... & Blaschke, T. (2020). Flood Susceptibility Mapping Using an Improved Analytic Network Process with Statistical Models. *Geomatics, Natural Hazards and Risk*, 11(1), 2282-2314.
- Yoo, H. J., Kim, D. H., Kwon, H. H., & Lee, S. O. (2020). Data Driven Water Surface Elevation Forecasting Model with Hybrid Activation Function: A Case Study for Hangang River, South Korea. *Applied Sciences*, 10(4), 1424.

- Yu, C., Gu, Z., Li, D., Wang, G., Wang, A., & Li, E. (2022). STSC-SNN: Spatio-Temporal Synaptic Connection with Temporal Convolution and Attention for Spiking Neural Networks. *Frontiers in Neuroscience*, 16, 1079357.
- Yu, P. S., Yang, T. C., Chen, S. Y., Kuo, C. M., & Tseng, H. W. (2017). Comparison of Random Forests and Support Vector Machine for Real-Time Radar-Derived Rainfall Forecasting. *Journal of Hydrology*, 552, 92-104.
- Yusoff, I. M., Ramli, A., Alkasirah, N. A. M., & Nasir, N. M. (2018). Exploring the Managing of Flood Disaster: A Malaysian Perspective. *Geografia*, 14(3), 24-36.
- Zaytar, M. A., & El Amrani, C. (2016). Sequence to Sequence Weather Forecasting with Long Short-Term Memory Recurrent Neural Networks. *International Journal of Computer Applications*, 143(11), 7-11.
- Zazo, R., Lozano-Diez, A., Gonzalez-Dominguez, J., T. Toledano, D., & Gonzalez-Rodriguez, J. (2016). Language Identification in Short Utterances Using Long Short-Term Memory (LSTM) Recurrent Neural Networks. *PloS One*, 11(1), e0146917.
- Zhang, J., Zhu, Y., Zhang, X., Ye, M., & Yang, J. (2018). Developing a Long Short-Term Memory (LSTM) based Model for Predicting Water Table Depth in Agricultural Areas. *Journal of Hydrology*, 561, 918-929.
- Zhang, Q., Wang, H., Dong, J., Zhong, G., & Sun, X. (2017). Prediction of Sea Surface Temperature Using Long Short-Term Memory. *IEEE Geoscience and Remote Sensing Letters*, 14(10), 1745-1749.
- Zhong, C., Guo, T., Jiang, Z., Liu, X., & Chu, X. (2017). A Hybrid Model for Water Level Forecasting: A Case Study of Wuhan Station. In *Proceedings of 2017 IEEE 4th International Conference on Transportation Information and Safety (ICTIS)*, (pp. 247-251).

- Zhou, Q., Teng, S., Situ, Z., Liao, X., Feng, J., Chen, G., ... & Lu, Z. (2023). A Deep Learning Technique based Data Driven Model for Accurate and Rapid Flood Predictions in Temporal and Spatial Dimensions. *Hydrology and Earth System Sciences*, 27(9), 1791-1808.
- Zhuang, W. Y., & Ding, W. (2016). Long Lead Prediction of Extreme Precipitation Cluster via a Spatiotemporal Convolutional Neural Network. In *Proceedings of the 6th International Workshop on Climate Informatics: CI*, (pp 1-4).

APPENDICES

Appendix A: Journal Publications

1. **Tei, R., & Saleh, A. Y.** (2022). Flood Prediction using Deep Spiking Neural Network.
International Journal of Circuits, Systems and Signal Processing, 16, 1045-1054.