AI-BASED HOTSPOT PREDICTION AND MANAGEMENT IN MULTICORE PROCESSORS

OBJECTIVES:

- Real-time Monitoring System
- Predictive Analytics Implementation
- Hotspot Detection
- System Stability and Reliability
- User-Friendly Interface
- Adaptability
- Data Visualization
- Anomaly Detection
- Performance Evaluation
- Pattern Analysis

INTRODUCTION:

- System monitors provide real-time information about CPU usage, temperature, memory usage, disk activity, and network activity.

- The primary function of a system monitor is to display the current state of the system.

- However, traditional system monitors only provide a snapshot of the system at a given moment. They lack the ability to predict future system behavior or identify potential hotspots before they occur.

INTRODUCTION:

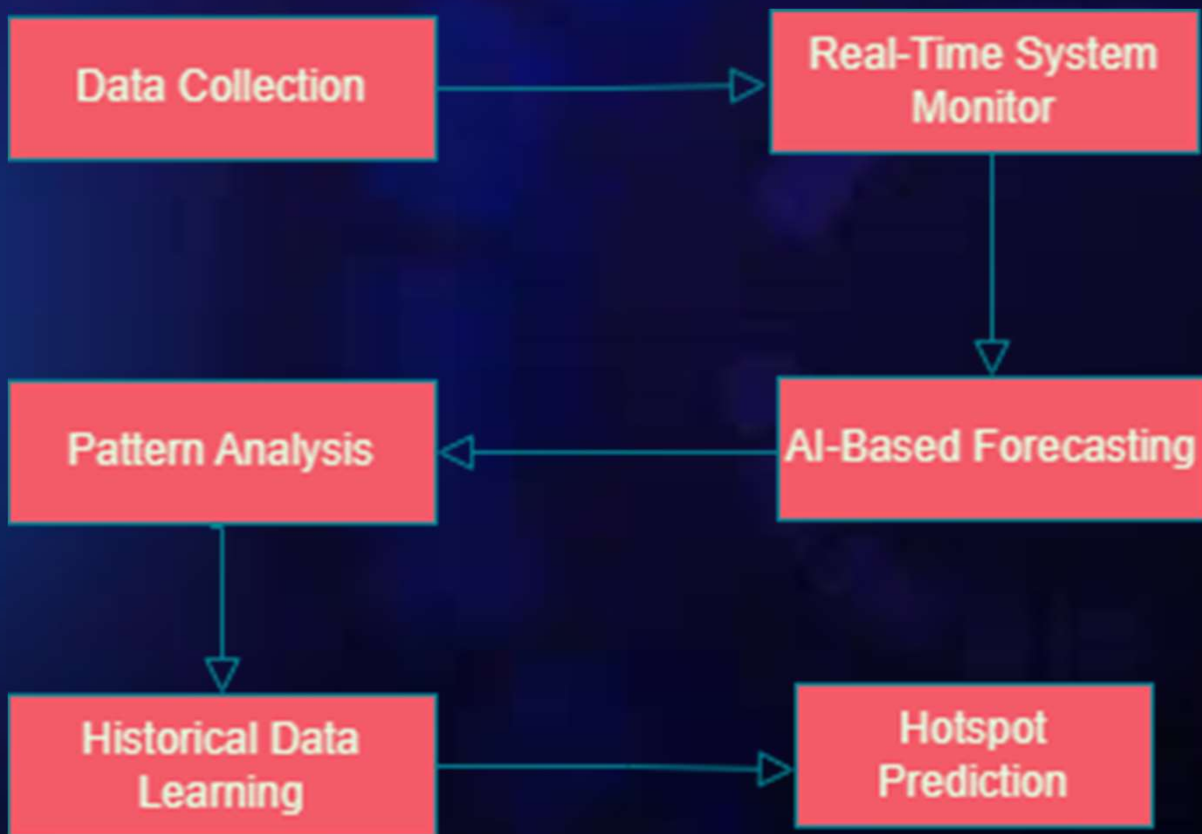
- AI-based techniques, such as predictive analytics and machine learning algorithms, can significantly enhance the capabilities of system monitors.
- These techniques can analyze patterns in system behavior and workload to forecast potential hotspots in multicore processors. Machine learning models can learn from historical data, enabling them to make accurate predictions based on specific workloads and usage patterns.
- By integrating AI with system monitoring, we can create a powerful tool that not only displays the current state of the system but also predicts future states.

LITERATURE SURVEY:

S.NO	TITLE	PUBLISHED YEAR	PROPOSED WORK	AUTHOR
1.	Hot-spot aware thermoelectric array-based cooling for multicore processors	2023	TEC-Array Cooling	Jinwei Zhang et al.
2.	Real-Time Full Chip Thermal Tracking: A Post-Silicon, Machine Learning Perspective	2022	Post-Silicon Approach for Thermal Tracking	Sheriff Sadiqbatcha et al.
3.	Algorithms for optimal thermal maps extraction and sensor placement on multicore processors	2022	Algorithms for Prediction of Hotspot	J. Ranier et al.
4.	Hotspot aware task mapping for lifetime improvement of multicore system using deep reinforcement learning	2021	Improving efficiency of the processor using deep reinforcement learning	Y. Gao et al.
5.	Hotspot identification and system parameterized thermal modelling for multi-core processors through infrared thermal imaging	2019	Hotspot identification using infrared imaging	H. Zhao et al.

PROPOSED WORK

BLOCK DIAGRAM



PROPOSED WORK

ALGORITHM

Step 1: Creating the Dataset

- Begin by setting up a system to collect temperature data from different cores in a processor. This could involve using software that can read the temperature of each core.

- Run a variety of software applications on the system that are known to create varying levels of heat. Collect this data over a period of 3 hours, taking a reading every second.

- Organize this data into an Excel sheet, with each row representing a different time point and each column representing a different core.

PROPOSED WORK

ALGORITHM

Step 2: Training the Model

- Load the dataset into a Python environment using the pandas library. This will allow you to manipulate and process the data easily.

- Preprocess the data using the MinMaxScaler from the sklearn.preprocessing module. This scales the data to a range of 0 to 1.

- Splitting the dataset into a training set and a testing set using the train_test_split function from the sklearn.model_selection module. A common split is 80% of the data for training and 20% for testing.

PROPOSED WORK

ALGORITHM

Step 2: Training the Model

- By Defining a machine learning model using the Sequential model from the tensorflow.keras.models module. This model should include LSTM (Long Short-Term Memory) layers, which are a type of recurrent neural network that can process sequential data, Dense layers for performing complex transformations on the data, and Dropout layers for preventing overfitting.

- Train the LSTM model on the training data. Monitor the model’s performance on the testing data throughout the training process.

PROPOSED WORK

ALGORITHM

Step 3: Prediction using the Trained LSTM Model

- Once the LSTM model has been trained, it can be used to make predictions on new data. This involves feeding new data into the model and having it output its predictions.

- These predictions represent the model's best guess at where hotspots will occur in the future based on the patterns it has learned from the historical data.

TOOLS USED

HARDWARE	SOFTWARE
PROCESSOR: Intel(R) Core(TM) i5-7300U CPU @ 2.60GHz 2.71 GHz.	**DATA MONITORING TOOL**: OPEN HARDWARE MONITOR.
SYSTEM TYPE: 64-bit operating system, x64-based processor.	**PROGRAMMING LANGUAGE**: Python.
	MODULES USED: PANDAS, NUMPY, SKLEARN, TENSORFLOW.

CONCLUSION

REFERENCE ARTICLES

1). S.X.-D. Tan, M. Tahoori, T. Kim, S. Wang, Z. Sun, S. Kiamehr, VLSI Systems LongTerm Reliability – Modeling, Simulation and Optimization, Springer Publishing, 2019.

2). J. Dousti, M. Pedram, Power-aware deployment and control of forcedconvection and thermoelectric coolers, in: 2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC), 2014, pp. 1–6, http://dx.doi.org/10.1145/2593069.2593186.

3). S. Jayakumar, S. Reda, Making sense of thermoelectrics for processor thermal management and energy harvesting, in: 2015 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED), 2015, pp. 31–36, http://dx.doi.org/10.1109/ISLPED.2015.7273486.

REFERENCE ARTICLES

4). J. Long, S. Ogrenci Memik, M. Grayson, Optimization of an on-chip active cooling system based on thin-film thermoelectric coolers, in: 2010 Design, Automation Test in Europe Conference Exhibition (DATE 2010), 2010, pp. 117–122, http: //dx.doi.org/10.1109/DATE.2010.5457225.

5). M.J. Dousti, M. Pedram, Power-efficient control of thermoelectric coolers considering distributed hot spots, in: 2015 Design, Automation Test in Europe Conference Exhibition (DATE), 2015, pp. 966–971, http://dx.doi.org/10.7873/DATE.2015.0517

6). D.M. Rowe, Thermoelectrics Handbook: Macro To Nano, CRC Press, Boca Raton, FL, USA, 2005