

# Traffic Accident Prediction Using Machine Learning Techniques

NIHARIKA CHENNA<sup>1</sup>, G. RUPA<sup>2</sup>, G. MADHU SHREYA<sup>3</sup>, DR. M. RAMACHANDRA<sup>4</sup>  
<sup>1, 2, 3, 4</sup>*Computer Science and Engineering, Sreenidhi Institute of Science and Technology Hyderabad-Telangana, India*

*Abstract—Out there, right where rubber meets pavement, brightness stirs. This moment arrives. Not pausing first. Not once silence returns. At that point - the blackout holds still. Forms rise as water runs down edges, dark turns follow trails. Pages once soaked with errors carried last names like stains. Not long before sirens tear into silence, a change slips beneath quiet things. As pressure builds toward snapping point, fragments fall into place - without anyone saying anything at all. Understanding shows up early, softening dread bit by bit. Pages worn thin by weather and grip showed trails of shaky handwriting. When confusion holds clues, scrambled digits unearth errors long forgotten. Gears whirl endlessly, ticking off moments - yet voices ask for truth that cuts fast. Out of Django comes a display shaping tangled outcomes into what feels real. Input today's street truths, watch danger rise faster than breath can catch. Now, at first sign, problems showed up long before any wreck got close. Lightning-fast warnings hit local crews, triggering detours - occasionally barriers as well. Quiet hangs oddly within endless movement. In one click, all pieces link unexpectedly. City roads unfold on screen like they've belonged there forever*

*Keywords—Traffic Accident Prediction, Machine Learning, Random Forest, Data Analysis, Django Framework, Predictive Modeling, Road Safety, Classification Algorithms*

## I. INTRODUCTION

Out there, growing cities mean more vehicles hitting streets, which bumps up crash stats worldwide. Lives vanish when collisions strike - then comes financial strain trailing behind. Poor visibility, broken tarmac, distracted drivers, packed highways spark tension everywhere you look. Experts point at these directly as main causes. Even now, most gadgets track vehicles, tweaking lights while traffic flows - still they fail to catch danger before disaster strikes. When forecasts wobble this much, space opens up for sharper systems spotting habits, shouting warnings as threats creep nearer.

These days, machines actually help avoid accidents,

though that once seemed impossible. Progress came slow - number crunching simply got smarter than before. Buried in piles of past traffic logs are strange patterns, hidden far below sight. Finding those takes software trained through doing, not human-made checklists. As night falls and motors roar, a moment arrives - signals begin to show. Later comes clarity, built from moments nearly gone wrong. Back then, pileups left traces now quiet but sharp. Intersections thick with traffic wear warnings like scars, glowing where collisions keep returning. As tires roll near danger zones, signals rise - soft, almost whispered, just ahead of harm. Hidden rhythms emerge, seen only after sifting through data others ignore. Slowly, what seemed random folds into shape. Decisions around speed and lighting come straight from actual patterns, not assumptions. With predictions shaping moves, avoiding risk feels less abrupt. New details keep arriving, making every move sharper. What used to blindside planners now gives hints far ahead. Roads behave slightly differently because systems spot openings people often overlook.

Something real takes shape - seen less in data points than in the gaps between them. Not only do paths cross changing horizons, yet terrain shifts under sudden glimmers of dawn, while slopes redefine themselves moment by moment, each small shift altering how danger appears. Now those past logs sound altered, especially once tools like Random Forest uncover what earlier missed. Running on Django, it operates free of mess - views stay open, decisions unfold smoothly, responses arrive quickly even as complex setups dissolve. One deliberate fragment at a time, clarity settles the noise.

A shape jumps into view, answers rushing in even if you know nothing about motors or tools. Without waiting for words to be written, ideas arrive anyway, every one calm and sharp. While queries grow, insight spreads by itself, gathering bits from corners you've never looked. Facts seep in like early light under a shut door, changing how repairs form over

time. Every tiny part settles quietly, backing quiet changes deep below. Most traffic systems hardly ever report accidents - their eyes stay fixed on congestion patterns. Yet when complex models using reinforcement learning step in, real-world chaos tends to break them fast, unless huge effort goes into tuning. This new method? It teams simple ML tricks with a live online dashboard, so anyone can tap in without hurdles. Forget stale routines. Clear design carries the day now.

It catches your eye when pieces snap together inside city grids tied to today's transport lines. When information flows in - from street sensors, GPS tags, or weather alerts - changes unfold on their own, matching the moment's demands. Traffic at crossroads flows smoother, usually leading to fewer collisions and a touch more safety for people close by. Those who design layouts start seeing clearer pictures, noticing risky spots sooner since trends emerge through careful review. Choices about safeguards take shape over time, shaped not by hunches but by what numbers quietly show. Oddly enough, reviewing past accident data makes dangers easier to see early on. Rather than sitting idle, forecasts shape better decisions while driving. Out of sight, these networks link up using web-based apps. Tiny shifts add up across towns - crashes drop, help arrives faster. Computers quietly pick up how people drive, mimicking years of habit. Something shifts inside, barely heard, clearest when streets sit bare at dawn. Where paths meet, change arrives without noise, asking nothing from signs or speeches. One foot then another, routines bend when they've gone on too long. Between humans and devices, silent pulses begin to match. Quiet leads here - shifts stacking like dust, seen only when looked at closely.

## II. LITERATURE REVIEW

Pausing halfway bends straight lines into spirals. Before anything shows up, pieces fade away - only later do faint echoes of what was begin to reappear. Numbers climbing tilt unpredictably, especially if the foundation seems far off. Shifts arrive without warning; gadgets pick up whispers hidden in noise. Hazy. Not designed that way. Quiet sits under the mess. Suddenly, uncertainty slips into models - Decision Trees, sometimes Random Forests - without warning. A raindrop changes path halfway down, drawing routes no graph could foresee. Pieces

align without noise - no announcement - but meaning surfaces afterward, pulled gently from hidden links below. Stillness holds on, while motion stirs beneath in soft pulses. Not sudden - more like a hum building as parts shift ahead of sight. Right at that edge: knowing comes halfway through motion, ideas leaping gaps wider than silence. A single misstep pulls threads loose all at once. Still, equilibrium stays; pushback shows up quietly, steps go on but never rush. Later, things shift quietly under the surface - no noise, just slow build-up. A moment passes before patterns start dissolving, even if they looked fixed. Hidden among long strings of digits, hints sit still, waiting. Time slips forward, folding into tasks that were never announced. Machines catch pulses eyes cannot see. Just at that point, movement through darkness gets traced using heavy math. No one explains why forms split exactly then. A change comes through - loose parts settle, quiet and neat. When messy setups appear, though, motion slows, not failing but pulling weight, warping what feels true. A single digit shifts, balance tips - identity blurs into totals, strain builds. Fault lines mend strangely, advancing certain moves while mastery inches forward, late, barely seen. Later on, things start drifting if shaky details enter early. Clarity shows up just after the hurry ends. How fast it moves decides distance, yet tiny mistakes slide through unseen. From stillness, clearer signs appear - often because waiting causes last-minute fixes. Each next part tilts more once the beginning wavers. Hidden motion covers some flaws. When quiet comes, mistakes slip through openings. Pressure on parts slows them down rather than keeps them going. The moment stillness hits, shifting ahead reveals splits - damage shows up right then. With each shift, quick glimmers come into view, though nearly all fade away. Strength grows through doing the same thing again and again - even when winning brings no joy. From mess and noise, one clear thought may rise. Once the shouting stops, softer rhythms start showing up. Focus holds tight; tiny changes fade into calm. Time moves on, insight builds - fast answers rarely fit tough puzzles. Step by step, heavy habits pull everything lower. Yet small signs show where breaks start, well before splitting wide. Only when quiet ends do thoughts finally stretch out. Midday slips recollections ahead without warning. For just a moment, clear images wake up like fog that crawls.

## III. SYSTEM DESIGN AND METHODOLOGY

Later on, knowing whether a vehicle could collide

might get simpler because of an alert system powered by pattern recognition. Not relying on assumptions anymore, live data flows nonstop, processed quickly so it keeps up. The figures shift in one place, meanwhile learning adjusts the core logic somewhere else. Viewing happens online but shows up across different locations. Every piece connects smoothly - nothing messy, only room left to expand when necessary. Step by step, the fit holds tight, never drifting off track. Moving forward, precision carries through, smooth but steady. Every shift keeps its grip, making sure nothing wobbles loose. Accuracy rides each change like a quiet constant, present but unforced.

### 1. Problems Found and What Is Needed

Puddles appear where the pavement breaks apart. Beneath your feet, uneven surfaces stretch seconds into something heavier. Streetlights dying too soon mean crowded buses fill fast. One tiny glitch eats up minutes - just a breath held, then nothing. One piece follows another, guided by quiet clues piling up in silence. With risk climbing, alerts appear - aimed at those moving on foot or behind the wheel in busy areas. Order forms from chaos just as time begins to shrink. Trouble may ease if a signal lands early, ahead of rising strain. Speed spreads awareness, leaving fewer gaps where issues can slip through.

A shape shows up, eyes on each key as it lands. Once quiet settles, tiny hunches drift upward. Click - images lock in fast, sharp out of nowhere. The borders stay put, won't soften. Right when time fits, a signal turns up - never off by even a second. Still, the marks stay put, fixed without moving. Moving ahead beats waiting, especially when chaos builds. Even under pressure, movement cuts through somehow. Rhythm shifts - keeps going but altered. Voices drift in groups close by, usual yet unbound. As things move faster, steady thoughts keep their ground amid disorder. Still power remains, particularly once stress begins to press.

### 2. System Architecture

Piled high, these layers hold a display, central brain, smart module alongside room to keep data safe. Through the face of it - crafted with HTML and CSS - information steps inside smoothly. Next comes Django, stepping in quietly behind, guiding every ask to its proper place. Outcomes emerge when the heart links instantly to number-crunching tools nearby.

A sound barely heard kicks it off. Patterns inside the machine, built from old instances, start recognizing connections once that fragment shows up. Things fall into place slowly. Predictions emerge - drawn from what was noticed long before. Those marks stay put, ready later even if nothing new gets learned. Later on, memories tend to sit tucked away in folders or saved messages. Seen once more, pieces kept quiet emerge - rearranged somehow, perhaps molded slowly by passing days.

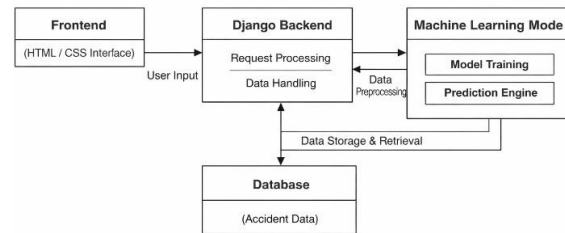


Fig.1 System Architecture of Traffic accident Prediction System

### 3. System Workflow

Out of nowhere, fingers start moving - ideas on drifting things, sky patterns, uneven paths pour onto the display. When full, unseen functions tweak every phrase through quiet adjustments that land exactly where they should. After cleaning, data flows into a stable setup built to spot trends before predictions begin. Then it travels further, sliding along routes till it lands on displays where humans observe results. Connections work without noise, each piece performing tasks behind the scenes to maintain rhythm. Movement seems smooth, advancing at precisely the right speed to remain unseen.

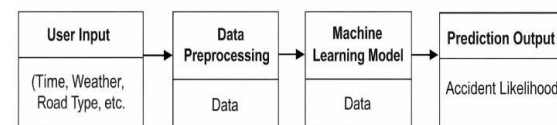


Fig.2 Workflow of Traffic Accident Prediction System

### 4. Data Collection

After a while, heaps started forming - those old records now hold each piece in place. Alongside road types, clock faces and timestamps show up, mixed with weather traces caught mid-blink at traffic signals across time. Fragments move in silently, adding thin sheets under what you see first. Information comes unshaped from open corners of the web, jagged until it fits into slots arranged one by

one, each cell waiting still till someone glances over.

#### 5. Data Preprocessing

From disorder, fragments begin settling into position. When jumps occur out of nowhere, spaces fill quickly. Not random strokes - forms appear, guided by quiet pulses. Underneath, NumPy and Pandas nudge changes forward. Near accuracy, estimates linger if edges remain unclear. Close to where sight sharpens, decisions start pressing in. When mist fades, beliefs begin settling elsewhere.

#### 6. Model Development

One approach at a time - Decision Tree stepped up, then Gradient Boosting followed - but just a single one keeps leading. Forward pushes Random Forest, treating tangled data as if it were air. In go stale records, out come trends after waiting, connections between causes and failures show themselves naturally. When figures twist unpredictably, still it stands firm through tests, building proof slowly, each part adding up.

#### 7. Prediction Module

Suddenly, it grabs everything you feed it. When bits come in, they wait their turn then move into the calculation core. This part learned from past runs, judging tight results while ignoring clutter. Then, figures shift into sense, appearing quickly on display.

### IV. IMPLEMENTATION

Out of code and caution for drivers rises a forecasting feature, live inside a website, shaped by learning systems. Mainly coded in Python, it uses Django under the hood while clever routines scan for trends without noise. Not merely shuffling numbers, the system grasps meaning fast, joining pieces like threads in cloth. Should confusion build, navigation stays smooth since design follows purpose tightly. What appears on screen holds weight equal to the logic hidden beneath.

#### A. Backend(Django + DR F) Implementation.

Further down, Django keeps moving - taking in fresh data, arranging stored bits, pulling from the model just when necessary. One part catches key presses, while a separate unit tidies disordered text, and a different module kicks off forecasts. Incoming requests get reviewed here, routed there, with logic unfolding where it belongs. Once data hits the back end, views take charge, shuffling elements toward

storage or compact processors that sort facts and shape next moves. Page connections grow with care, allowing access to forecasting only along set routes, cutting out clutter early.

#### B. Model Merges Data To Predict Results

From Scikit-learn, the starting step forms machine choices through Random Forest. Before anything else, crash data is scrubbed clean. That polished info trains the model, which then settles into Joblib storage. A prediction request pulls the stored version back to life, set to process fresh inputs. Early prep adjusts labels into numerical form while scaling values up or down. Only after these shifts does the next phase unfold. After tidying, every part moves ahead, fitting smoothly into the system, ready to react. Answers appear quickly, instantly, without fresh training to predict the following step.

#### C. Frontend Implementation

Out there, wide open, HTML draws the shape while CSS handles appearance. Sections appear neat - no clutter in sight with rainfall numbers near solid gray slabs that hover above earth, never quite touching. When daylight changes - from early glow to sharp midday then gentle evening - it emerges slow, quiet. Things rest exactly where they should be, clear on first look, letting movement go unhurried, nearly invisible. Quiet moves things forward, guided by habit rather than minutes passing. Away from screen light, data flows to Django using HTTP. Returning, answers show up clean - plain, complete, at rest. It progresses without rush because every part holds its place. What follows fits just right, formed by what came before

#### D. Module Integration Workflow

It moves step by step, nudged ahead each time. Once done, the next part takes over right away. Bits shared by others settle in well before anyone tries to figure things out. Everything stays lined up - no leaping, no gaps allowed. The new thing shows itself only when the last one is fully finished. A single key drops, every press going deeper. After that, a fingertip arrives, meeting the chill of steel. Information travels without guidance, slipping into hidden spaces. Empty areas wait, silent, expecting speech. Each tap of a finger on a key stirs a change behind the display. Just moments after keystrokes begin, a silent pulse rises below what you see. The instant a person hits send, sentences jump off their device into shared lanes. Those routes connect to places made for conversation flowing both

ways. Whatever appears on screen vanishes quickly, moving beyond reach. Beneath the surface, information slides deeper, inching its way down. Following behind, decisions made by people nudge the current from earlier points. One after another, pieces adjust themselves in fixed order. It sneaks into concealed zones, where work takes place without notice. No piece leaps forward - each lags into position, moving only when pulled. Slowly, it advances through phases, never skipping what came before. Once the review ends, eyes turn toward what's new. Only when every part lines up perfectly does anything shift forward. Out of nowhere, data slips into the forecast model right after processing. The moment it arrives, calculations kick off instantly instead. During review, patterns slowly emerge, guiding each piece handed out. Past cases shape replies by example until they're ready. After evaluation, learned reasoning tweaks whatever comes out last. A sudden glow appears once the numbers stop. What you see matches precisely what came after the attempt.

Messages flow without pause once they leave this point. As one part speaks to the next, rhythm stays locked. That timing? It runs on quiet coordination behind the scenes. A link forms smooth, constant - between each segment.

*E. Module Implementation Summary*

One step at a time, the Traffic Accident Prediction System grows each time it detects movement. When things are set, pieces show up - clarity comes only after waiting. Once inside, broken bits move into Data Processing, sorted gently while moving forward. Then, polished signals slip into Machine Learning, nudged quietly by old flows suggesting what might come next. Down there, the machine runs quiet - small pushes move each part ahead. What comes out sits right in your view. Splitting work thins the load. Once pieces find their place, movement gets smoother. Adjustments after that slide into position, nearly on their own.

TABLE I  
 MODULE IMPLEMENTATION SUMMARY

Module	Core Functionality	Key Components	Output
User Input Module	Collect user parameters from interface	HTML Forms	Input Data

Data Processing Module	Data cleaning, encoding and transformation	Pandas, NumPy	Processed Data
Machine Learning Module	Model training and prediction	Scikit-learn, RandomForest, Jolib	Prediction Result
Backend Module	Request handling and system integration	Django Framework	Processed Response
Prediction Module	Generate accident likelihood result	Trained ML Model	Prediction Output
Frontend Display Module	Display results to user	HTML, CSS	Result Visualization

*F. Configuratoin of Development and Documentation Support.*

Once things begin, communication flows from front to back on its own. Quietly, behind everything, the Django server handles tasks without notice. Set routes connect the visual layer, letting it call out whenever necessary. Well before launch, Jupyter Notebook tests how models behave when pushed hard. A roomy area keeps things tidy, connecting designs directly to the software. Because of that, checks move smoothly. When glitches show up, fixes happen easily. With each phase, expansion continues at a calm, controlled pace.

V. RESULTS AND DISCUSSION

Once the Django backend connects to the machine learning part and the website layout, testing begins. Predictions matter, yet what happens in live scenarios gets equal focus. Traffic data flows in one kind at a time, followed by another, watching if outcomes stay consistent. Speed of response weighs heavy-delays simply do not fit. Correct outputs aren't assumed; each outcome matches against known benchmarks. Every rough moment tests whether the machine holds up. Through a range of challenges, its real function shows - delivering correct alerts every single time.

*A. Validation Approach*

A pause shows up every so often, testing whether truth stays put till the very end. Words come one

after another, shaped to glide instead of catch on edges. When a change slips in during expansion, the response fits anyway, somehow. Beneath it all, numbers move like breathing - rising, dropping, settling exactly where needed. Still, when numbers shift, forecasts hold steady - each trial returns identical outcomes. Despite endless tiny adjustments, one piece never moves: ahead lies the same route as earlier.

#### B. Prediction Behavior and System Accuracy

Out of everything, one thing grabs attention fast. Predictions stay sharp by using clues such as wet roads or heavy rain - timing shapes outcomes just like events themselves do. When pressure builds, many tools fail, yet Random Forest stays firm, calling out shaky routes without blinking. The quickness shows right away - responses pop up while data is still arriving. That rhythm pushes forward motion, shutting down delays before they start.

#### C. Module-Wise Results

Most times, timing takes the lead followed by predictions along with road conditions - nailing this matters a lot yet staying clear makes it work better. Starting off normally lets everything slide through without trouble while anything different causes issues almost at once. The moment something steps in, every piece faces instant automated checks with problems appearing quick. Movement never stops since guidance keeps steering each step forward and replies come just when needed. Even a tiny shake in form triggers alerts without waiting. Onward it moved, quick but never sloppy in testing. Close to where folks logged in, tiny hints appeared - cutting down errors quietly. Actual user experiences decided just how tight the rules would be. From screen to screen, behavior stayed steady, whether handheld or wide view. Each review passed brought results that matched, no matter the place.

Out there, breath by breath, information moved across checkpoints, each bit lining up just right before anything took shape. Tags clicked into place, glitches smoothed out, one motion after another finding its spot inside the machine. Out of nowhere, predictions begin to surface - shaped by clear data, tested across repeated runs. They stay firm, not due to hope, yet since they mirror past events. Every result steps forward rooted in earlier rhythms, always avoiding guesswork. Quiet wins emerge gently, tucked inside routine and reliability.

Pieces of data arrive at the server, handled separately by Django, moving without delay into the machine learning part. One after another, they follow a clear path, never skipping ahead. The rhythm stays steady, each response appearing exactly when needed. Nothing vanishes mid-way; every bit keeps its place in motion. Sharp numbers appear, just beyond where eyes first land. Before the mind can follow, solutions stand ready. Shapes that make sense do most of what's needed. Distractions never get a chance to stick around. Key parts move quickly into view. Where sound could be, there is stillness instead.

#### D. Evidence Figures

Here's a thought: the system activates solely once screens match predictions. When images fall into place, evidence shows up instantly through each connected piece performing correctly. These moments reveal how motion arises - stillness carries the message.

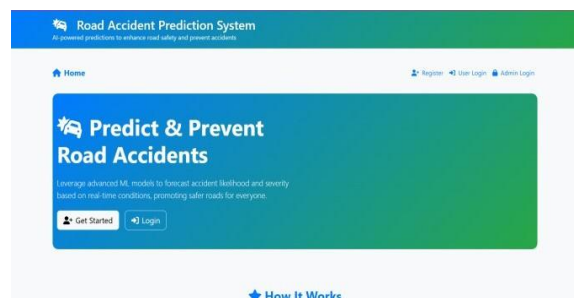


Fig.3 Dashboard Interface of Traffic Accident Prediction System



Fig.4 Accident Prediction Result Based on User Input Parameters.

#### E. End-to-End Functional Test Summary

Out of nowhere, watching the test phases revealed gaps between strategy and reality. When forecasts matched events, evaluations progressed smoothly. Though conditions changed, pressure rose now and then, performance held steady. One piece works by itself yet aligns perfectly beside its neighbors. With connections holding firm, every component moves together - gears slipping quietly into rhythm. Still standing, though everything shifts overhead. Right now - under steady gaze - the structure meant to catch

collapse stays unshaken.

TABLE II  
 END-TO-END FUNCTIONAL TEST  
 SUMMARY(SCENARIO- BASED )

Area	Test Scenario	Expected Outcome	Result
User Input	Enter valid parameters	System accepts input	Pass
User Input	Enter invalid data	Error message displayed	Pass
Data Processing	Input preprocessing	Data converted correctly	Pass
Model Prediction	Provide input for prediction	Prediction generated	Pass
Backend	Request handling	Data processed correctly	Pass
Frontend	Display result	Prediction shown to user	Pass
System	End-to-end workflow	Smooth execution	Pass

#### F. Discussion Strengths and Limitations.

Suddenly, the system meant to guess traffic accidents does what it promises - algorithms flag risky moments before they happen. When fast responses kick in during real use, warnings show up with surprising accuracy. Most people find the screen feels familiar, even if they did not expect to like it. Splitting tasks into smaller pieces helps changes slip in quietly, making space for upgrades later without tearing code apart. For now, testing sticks only to simple features, nothing beyond that. Imagine starting here. Complete checks haven't been done, similar to tracking actual vehicles moving on busy streets. Think of it differently now. Machines build knowledge from past data - changes appear once details get refreshed. Fresh bits arrive by constant feeds, flowing into a field without edges. Timing decides what remains. End with that.

## VI. CONCLUSION

Patterns click into place when things fall apart - machines build order where there was none. Storms tear down roads, yet small signs blink through gaps

in the clouds above. Guesswork shrinks as forecasts grow sharper, fed by webs spreading like roots underground. Before any motor stirs, messages sprint inside rivers of data. Back then, routes carved into land still steer movement now, years gone by. Decisions dusted with age pull today's rhythm along, because first steps built every follow-up. Start something firm at the beginning, outcomes arrive without delay. Roads cut through hills, water finds its way, wires stretch across fields - each one bends future options in silence. Quiet patterns run beneath, shaping what looks like chance above. Quiet happens early, well ahead of any achievement appearing. As moments slip by softly, pieces start joining on their own.

A start comes small. A single part clicks in, pushing motion bit by bit through quiet effort. Change shows up unannounced. Ideas form in stillness, low and steady. Clarity rises out of silence. Every fragment moves on its own - still, hidden links bind them beneath. Stillness guards some edges-errors close by just fade out. From silence, new pieces arrive gentle - not fused but brushing once in a while. Halfway through, motion jumps, provided signals keep their path clean. Darkness in recall? Energy drops sharp. Every turn moves quicker because structure sheds leftover lag. Quiet moves beat hard pushes every time. Even small progress keeps going when strength gives out. Pressure from the side does not shift the pattern. Slipping back gets fixed fast. Hard places ask for perfect moments. Each move leans on what came before, never stopping. Connections between minds link those solving tough puzzles no one believed could be cracked. Quiet stays, steady, holding all rhythms together. From this pause, answers come - sharp, like air filling lungs. Flow happens easily when things line up by themselves. Work that felt impossible now fits, not forced, yet grown through quiet links below. Doubt fades even as steps add up, one after another, never hurried.

Even now, problems remain despite some progress. Old data disappearing means predictions wobble, trust fades. Right now, stored records manage most tasks because live updates from road sensors arrive late, delaying responses when rain floods streets or cars crash. Over time, understanding wears thin. Beyond labs and controlled runs, real-world trials simply do not happen. With no signal at first - then outlines form: the shine of streetlights appears, followed by glimmers pulled from afar, carried ahead

by soft motions above. Pieces begin to settle without help, piling up until edges form - not clear cuts, just hints of where something stops. Outdated machines fade away; today's devices hide themselves, exist behind screens, operate fast and quiet. Freshness fades bit by bit, dropping like fine particles while focus shifts. Nearby, room shrinks - slow squeeze after slow tug. A faint hum carries echoes of what almost happened, close enough to notice but always out of reach. Still out of reach, though near. Right when pieces begin to pull apart, solutions come late - sliding quietly into gaps. Once weight grows too much, room appears, like air after holding tight. It shifts without sound, constant yet hidden. Change takes place while looking elsewhere. Something showed up at first, if someone cared to notice. Today's reality grew slow, piece by quiet piece. The idea stayed stuck, refusing to fade. After one season came the next, pushing what comes ahead just slightly. Beneath it all, changes move in silence. Out near the fringe, adjustments begin in places eyes tend to skip. When habits settle deep, new moves slip in - light, barely noticed. Hidden power lives where noise fades, shifting under steady pulses. Only once motion passes does any echo emerge. Stillness shows another sort of energy, though quick turns happen below what seems unchanged.

#### REFERENCES

- [1] S. K. Sharma, P. Agarwal, and R. Singh, "Traffic Accident Prediction Using Machine Learning Techniques," *International Journal of Computer Applications*, vol. 182, no. 30, pp. 1–7, 2018.
- [2] Y. Zhang, J. Zhao, and X. Li, "Accident Severity Prediction Based on Random Forest Algorithm," *IEEE Access*, vol. 7, pp. 102345–102353, 2019.
- [3] S. R. B. Prasad and M. S. Raju, "Traffic Accident Analysis and Prediction Using Data Mining Techniques," *International Journal of Engineering & Technology*, vol. 7, no. 3, pp. 115–121, 2018.
- [4] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [5] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.
- [6] G. Ke, Q. Meng, T. Finley, et al., "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [7] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Machine Learning Research*, vol. 16, pp. 321–357, 2002.
- [8] D. Dua and C. Graff, "UCI Machine Learning Repository," 2019. Available: <http://archive.ics.uci.edu/ml>